



FirstSpirit™

Unlock Your Content

Manual for Administrators

FirstSpirit Version 5.1

Version	1.38
State	RELEASED
Date	2015-02-04
Department	FS-Core
Copyright	2015 e-Spirit AG
File name	ADMI_EN_FirstSpirit_AdminDocumentation

e-Spirit AG
Stockholmer Allee 24
44269 Dortmund | Germany

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

info@e-spirit.com
www.e-spirit.com

e-Spirit

Table of contents

1	Introduction.....	12
1.1	Topic of this documentation	17
1.2	Structure of this documentation	18
1.3	Supplementary documentation	19
2	System Requirements and Installation.....	20
3	FirstSpirit Server control	21
3.1	Unix.....	21
3.1.1	GNU/Linux and Solaris 9.....	21
3.1.2	Solaris	21
3.1.3	AIX.....	22
3.1.4	Via a normal user account.....	22
3.1.5	Generate a stack dump	22
3.2	Windows.....	23
3.2.1	Register / Deregister as system service via the start menu	23
3.2.2	Uninstall via the start menu	23
3.2.3	Start /Stop as a system service via the start menu.....	24
3.2.4	Start /Stop in console mode	24
3.2.5	Edit configuration files via the start menu	24
3.2.6	Further start menu functions	26
4	FirstSpirit Server configuration	28



4.1	File system organisation	28
4.1.1	File names	28
4.1.2	Web applications	29
4.1.3	Directory structures	29
4.2	General configuration information	30
4.3	Configuration files (FirstSpirit Server)	32
4.3.1	FirstSpirit Server configuration (fs-server.conf)	32
4.3.2	Configuration of the Java VM and the Java Wrapper (fs-wrapper.conf)	74
4.3.3	Database connection configuration (fs-database.conf)	84
4.3.4	Login process configuration (fs-jaas.conf)	85
4.3.5	Licence configuration (fs-license.conf)	101
4.3.6	Logging configuration (fs-logging.conf)	103
4.3.7	Web server configuration (fs-webapp.xml)	106
4.4	Connection to an LDAP server	110
4.4.1	Authentication via LDAP	110
4.4.2	Bind LDAP attributes to a FirstSpirit user	111
4.4.3	Use TLS or SSL	112
4.5	Integration into an external web server	113
4.5.1	Apache HTTP Server with the Jetty servlet engine	114
4.5.2	Apache HTTP Server with the Tomcat servlet engine	118
4.5.3	Tomcat servlet engine	121
4.5.4	Tomcat servlet engine on a dedicated host	129
4.5.5	External servlet engine and load balancing on multiple servlet engines	137
4.6	Integration into an external application server	138
4.6.1	Integration into WebSphere Application Server	139
4.6.2	Logging for FirstSpirit web applications	141



4.6.3	External application server requirements.....	141
4.7	HTTPS server configuration.....	143
4.7.1	Install a security certificate for a test server	143
4.7.2	Install a trusted security certificate	144
4.8	Additional security measures (FirstSpirit 5.1R4 and higher).....	148
4.8.1	Parameterizing encryption	148
4.8.2	Repository encryption.....	153
4.9	Database connection.....	155
4.9.1	Storing the JDBC driver files	155
4.9.2	Creating a JDBC driver module	156
4.9.3	Installation and configuration of the JDBC driver module	162
4.9.4	Data source configuration	168
4.9.5	Required permissions for database user accounts.....	175
4.9.6	Notifications and restrictions concerning the specific database systems	177
4.9.7	Examples for linking different database systems	179
4.9.8	Procedure for connecting external databases	183
4.10	Roll-out process for native applications.....	184
4.10.1	Roll-out process (server).....	184
4.10.2	Roll-out process (workstation computer).....	185
4.10.3	Updating the native system components	187
4.10.4	Preventing the overwriting of files during the roll-out process	188
4.10.5	Roll-out directory requirements	188
5	FirstSpirit web application configuration	189
5.1	FirstSpirit start page configuration (fs5root).....	189
5.2	ContentCreator configuration.....	189



5.2.1	Project prerequisites when using ContentCreator.....	190
5.2.2	ContentCreator as a local project application.....	191
5.2.3	Browser configuration when using ContentCreator	192
6	FirstSpirit start page.....	194
6.1	Automatic login using single sign-on (SSO).....	194
6.2	Login with user name and password.....	195
6.3	FirstSpirit start page	196
6.3.1	Starting applications	197
6.3.2	Quick start.....	198
6.3.3	User	199
6.4	Starting the applications	206
6.4.1	ContentCreator	206
6.4.2	SiteArchitect	206
6.4.3	ServerManager.....	208
6.4.4	ServerMonitoring	208
6.5	Starting FirstSpirit Client as a Java application.....	208
6.5.1	Socket mode	209
6.5.2	HTTP (Internet) mode	211
7	FirstSpirit ServerManager	213
7.1	Server and project administrators	213
7.2	Menu bar items	214
7.2.1	File.....	214
7.2.2	Server.....	215
7.2.3	Project.....	218
7.2.4	User	235



7.2.5	Extras	241
7.2.6	Help.....	242
7.3	Server properties	243
7.3.1	Global server properties	243
7.3.2	Presentation channels.....	247
7.3.3	Conversion rules.....	248
7.3.4	Installed fonts.....	250
7.3.5	Databases.....	252
7.3.6	Language templates	257
7.3.7	Webstart.....	260
7.3.8	Start page.....	261
7.3.9	Schedule overview, schedule management and action templates.....	263
7.3.10	JAAS configuration.....	264
7.3.11	Modules	265
7.3.12	Web server	271
7.3.13	Web applications	278
7.3.14	Clustering.....	284
7.4	Project properties	292
7.4.1	Project.....	293
7.4.2	Options.....	294
7.4.3	Substitutions.....	304
7.4.4	Fonts.....	306
7.4.5	Languages	307
7.4.6	Resolutions.....	310
7.4.7	Users	313
7.4.8	Groups	315
7.4.9	Schedule overview	322



7.4.10 Schedule management.....	323
7.4.11 Action templates	324
7.4.12 Databases.....	325
7.4.13 Template sets	326
7.4.14 ContentCreator settings.....	330
7.4.15 Quota.....	338
7.4.16 Permissions.....	339
7.4.17 Project components	341
7.4.18 Web components.....	343
7.4.19 Remote projects.....	350
7.4.20 Media constraints	354
7.4.21 Client applications	357
7.4.22 Repository.....	366
7.5 Schedule entry planning.....	370
7.5.1 Schedule overview	372
7.5.2 Schedule management.....	375
7.5.3 Action templates	378
7.5.4 Add/Edit schedule entry (Properties tab)	380
7.5.5 Add/Edit schedule entry (Actions tab).....	384
7.5.6 Adding actions to a schedule entry	386
7.5.7 Copying actions from a different schedule entry.....	389
7.5.8 Inserting actions using action templates	390
7.5.9 Server-based actions	391
7.5.10 Project-based actions.....	401
7.6 Clustering – load distribution on generation.....	422
7.6.1 Concept	423
7.6.2 Check licence file.....	424



7.6.3	Configuration of the cluster nodes.....	425
7.6.4	Configuration of the generation schedule	425
7.7	Configuration of the spelling check.....	428
7.7.1	Install/uninstall SpellService (server properties).....	428
7.7.2	Update SpellService (server properties)	430
7.7.3	Configure Global SpellService.....	431
7.7.4	Configure global dictionaries.....	433
7.7.5	Start and configure "SpellService" service	436
7.7.6	Add SpellService as project component.....	436
7.7.7	Project-specific SpellService configuration.....	438
7.7.8	Add project-specific dictionaries	440
7.8	Support for Apache FOP	441
7.9	Project archiving	443
7.9.1	Version history	443
7.9.2	Revisions.....	443
7.9.3	Minimum project archiving requirement.....	444
7.9.4	Version history after archiving	446
8	FirstSpirit ServerMonitoring.....	449
8.1	Overview	450
8.1.1	Overview – Status	450
8.1.2	Overview – Activities	451
8.1.3	Overview – Sessions.....	452
8.2	Projects.....	454
8.2.1	Projects – Overview	454
8.2.2	Projects – Statistics.....	455
8.3	Log files	458



8.3.1	Log files – complete server.....	458
8.3.2	Log files – by project.....	462
8.3.3	Log files – by deployment	464
8.4	Scheduling.....	466
8.4.1	Scheduling – Overview	466
8.4.2	Scheduling – Planned schedules	467
8.4.3	Scheduling – Executed schedules	467
8.5	Users.....	469
8.5.1	Users – Find.....	469
8.6	FirstSpirit.....	470
8.6.1	FirstSpirit – Configuration.....	470
8.6.2	FirstSpirit – Control	477
8.6.3	FirstSpirit – Message.....	485
8.6.4	FirstSpirit – Databases.....	485
8.6.5	FirstSpirit – Monitoring.....	486
8.6.6	FirstSpirit – Clustering.....	493
9	FirstSpirit JMX Console	494
9.1	Starting the JMX console	495
9.2	MBeans	496
9.3	BerkeleyDbBackend	499
9.3.1	Attributes	499
9.3.2	Operations	500
9.4	Cache Size Manager.....	501
9.4.1	Attributes (all).....	501
9.4.2	Operations	501
9.4.3	Attributes (project-based).....	502



9.5	Content Manager	502
9.6	Discovery Manager	503
9.6.1	Attributes	503
9.6.2	Operations	503
9.6.3	Attributes (server-based).....	503
9.7	Event Manager	504
9.8	ExecutionManager	505
9.8.1	Classification of thread queues	505
9.8.2	Processing within the ExecutionManager	506
9.8.3	Attributes	507
9.9	Lock Manager	509
9.10	Media Manager	510
9.11	Module Manager	510
9.12	NIO Socket Server	510
9.13	Preview Manager	511
9.14	Project Manager	512
9.14.1	Attributes	512
9.14.2	Operations	513
9.15	Reference Manager	514
9.15.1	Attributes	514
9.15.2	Operations	515
9.16	Registry Manager	515
9.17	Repository Manager	515
9.18	Search Manager	516
9.18.1	Attributes	516
9.18.2	Operations	517
9.19	Server Action Manager	517



9.19.1 Attributes	517
9.19.2 Operations	518
9.20 Server Manager	518
9.20.1 Attributes	518
9.20.2 Operations	518
9.21 Service Manager.....	519
9.21.1 Attributes	519
9.21.2 Operations	520
9.22 Session Manager.....	520
9.22.1 Attributes	520
9.22.2 Operations	521
10 Secure deployment via rsync and ssh.....	522
10.1 Web server under Unix.....	522
10.2 Web server under Windows	523
10.3 FirstSpirit Server under Unix.....	524
10.4 FirstSpirit Server under Windows.....	525
10.5 FirstSpirit project configuration.....	526
11 User permission configuration	528
11.1 Introduction.....	528
11.1.1 Define user permissions	529
11.1.2 Check user permissions	530
11.1.3 Assigning user permissions.....	531
11.2 Architecture	532
11.2.1 Introduction	532
11.2.2 Overview	533



11.2.3 Authorization checks using FirstSpirit	537
11.3 Protection of personalized project content in FirstSpirit	538
11.3.1 Personalization	538
11.3.2 Checking access rights via the Access Control Database.....	539
11.3.3 Secure Media concept	541
11.3.4 Scope	542
11.3.5 Permission definition	543
11.3.6 Definitions	546
11.4 Configuration.....	547
11.4.1 Introduction.....	547
11.4.2 Activating the permission service	547
11.4.3 Configuring the server component.....	547
11.4.4 Configuration for the deployment	552
11.4.5 Permission configuration via the project properties	553
11.5 Application in the LIVE system.....	553
11.5.1 Servlet server configuration.....	553
11.6 Application in the project.....	554
11.6.1 Manual group definition and ID assignment.....	554
11.6.2 Configuration semantics	556
12 Appendix: Configuration files	558
12.1 fs-wrapper.conf	558
12.2 fs-jaas.conf	562
12.3 fs-webapp.xml	563



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™ SiteArchitect or the FirstSpirit ContentCreator. 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:

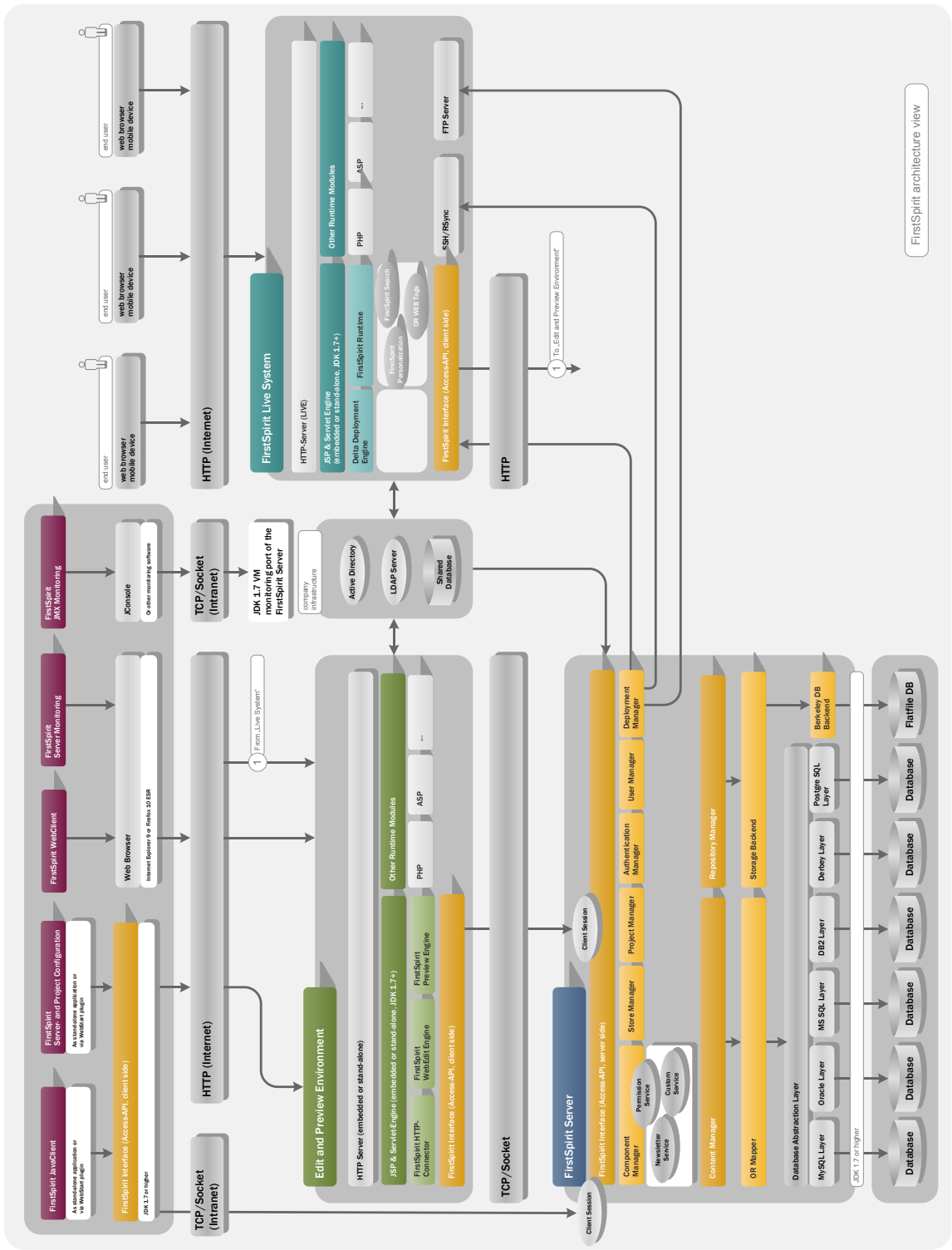


Figure 1-1: Total view



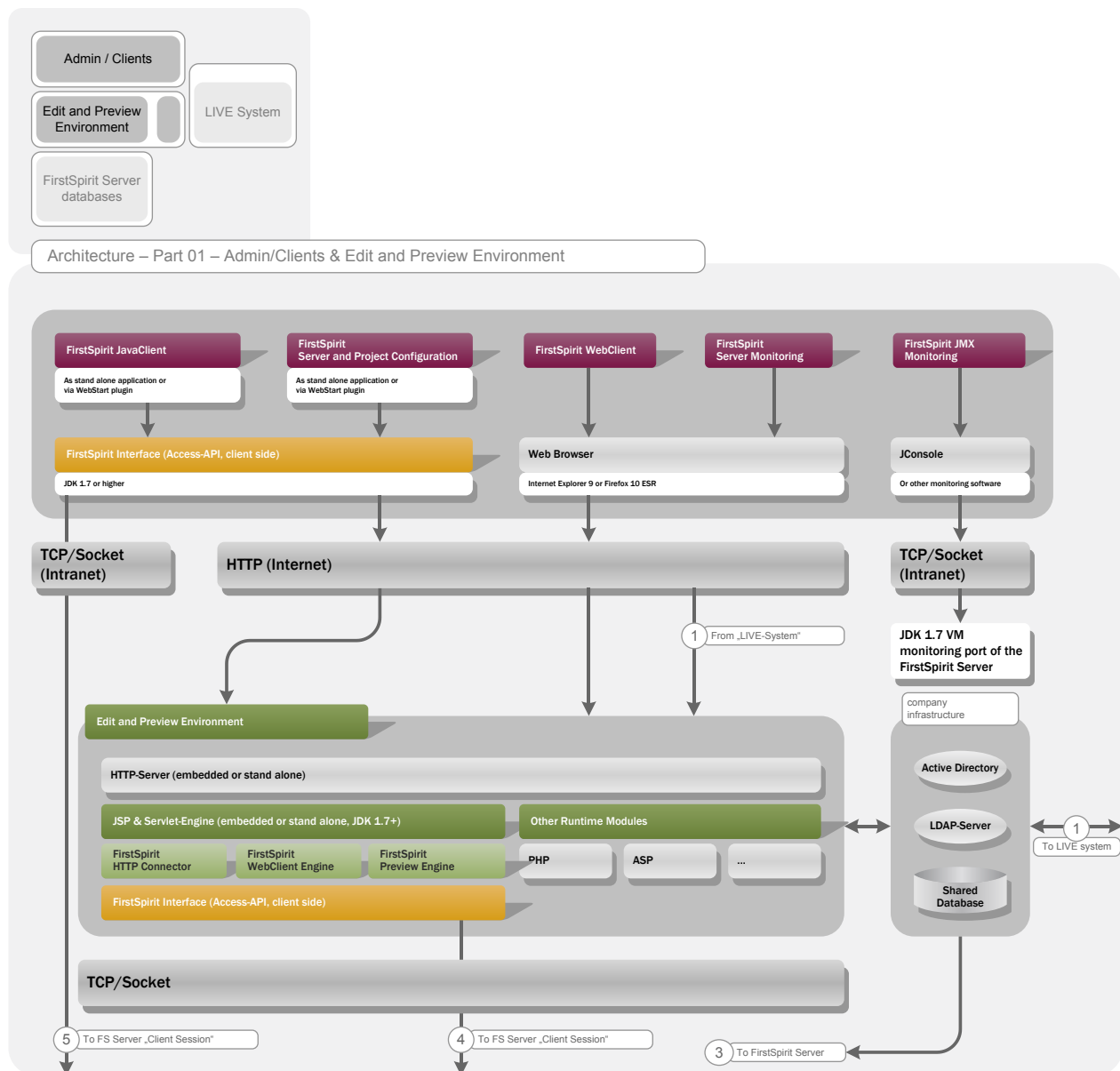


Figure 1-2: Total architecture – Detail: Edit and Preview Environment



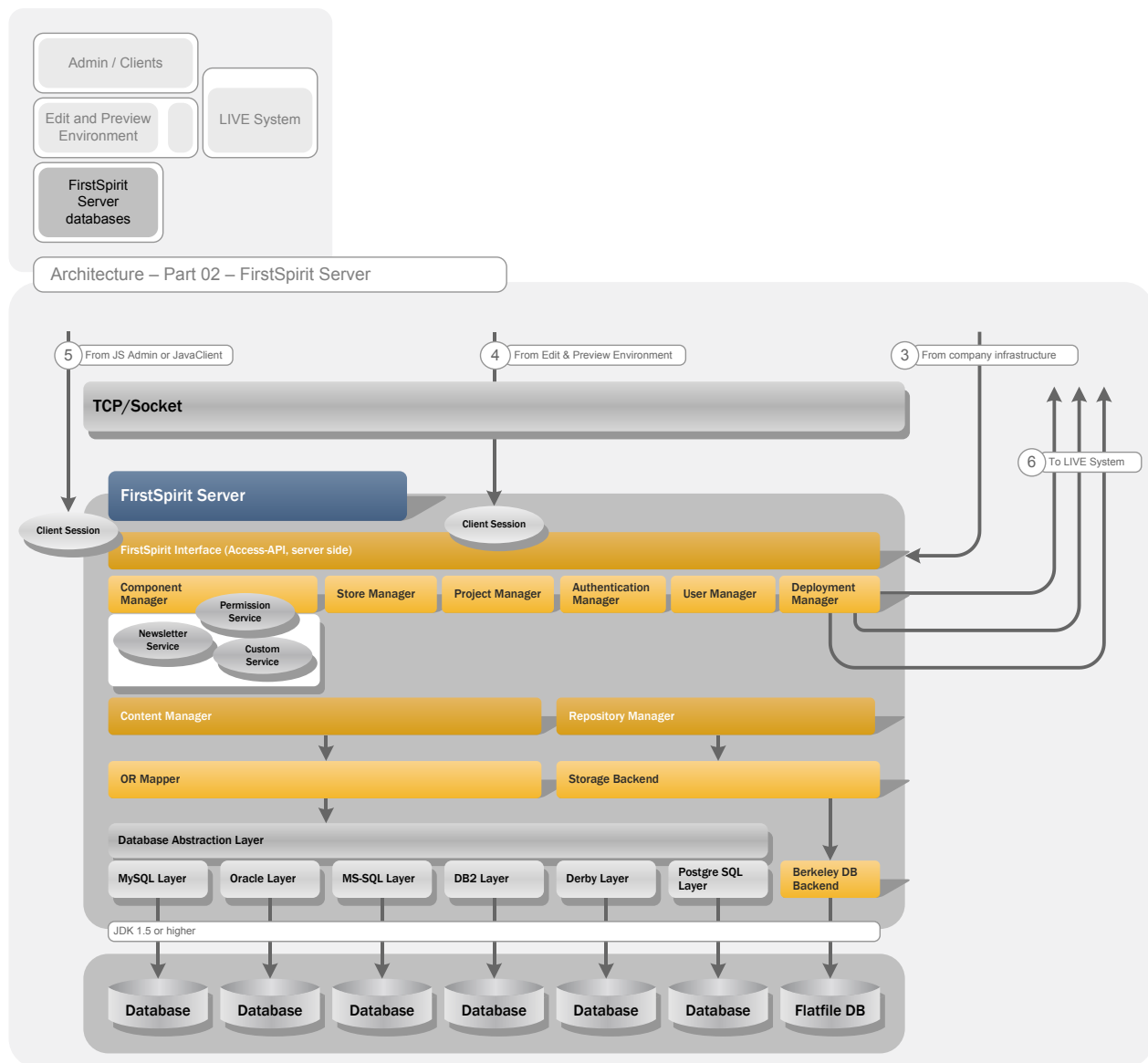


Figure 1-3: Total architecture – Detail: FirstSpirit Server



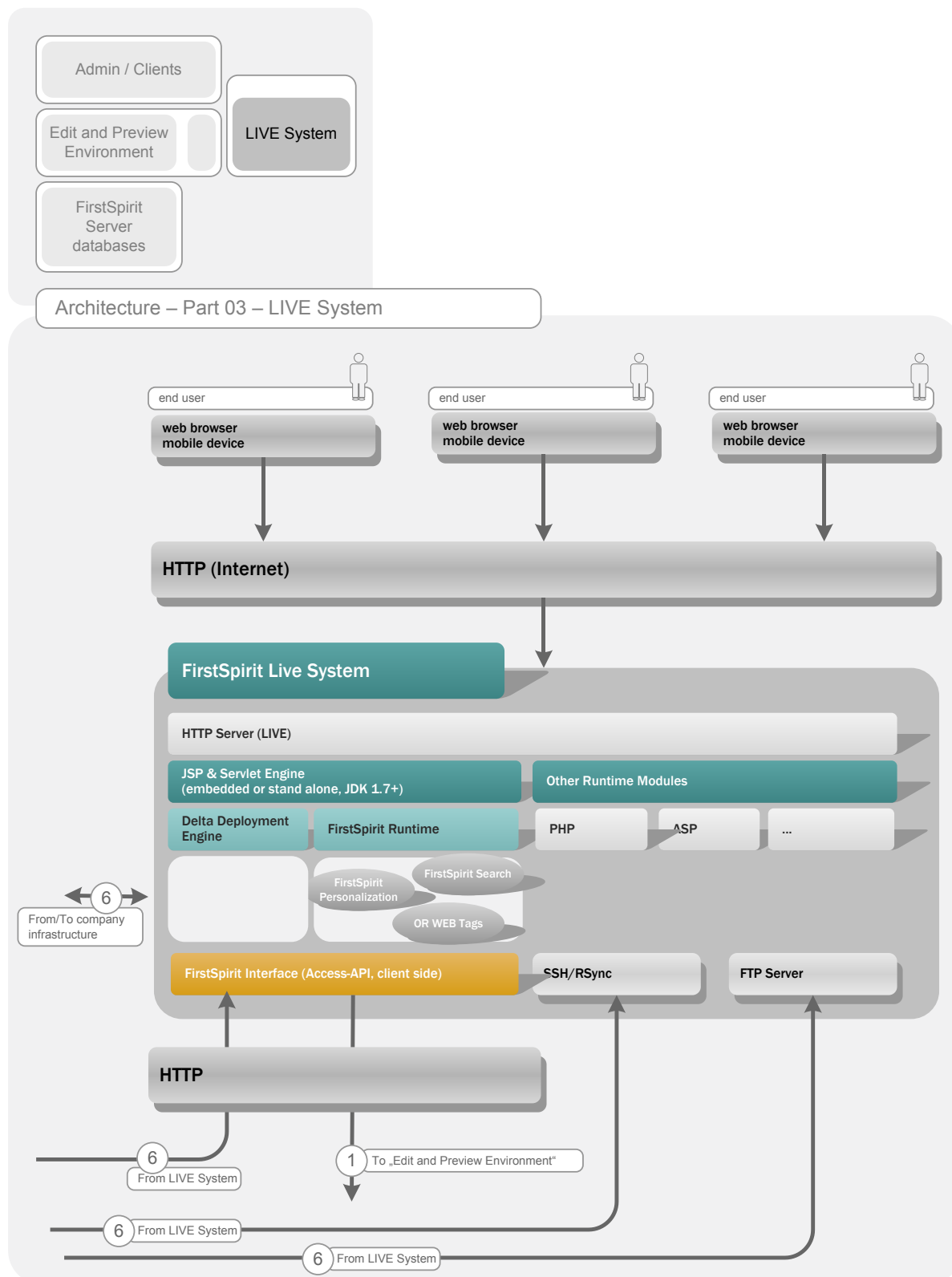


Figure 1-4: Total architecture – Detail: LIVE system



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.

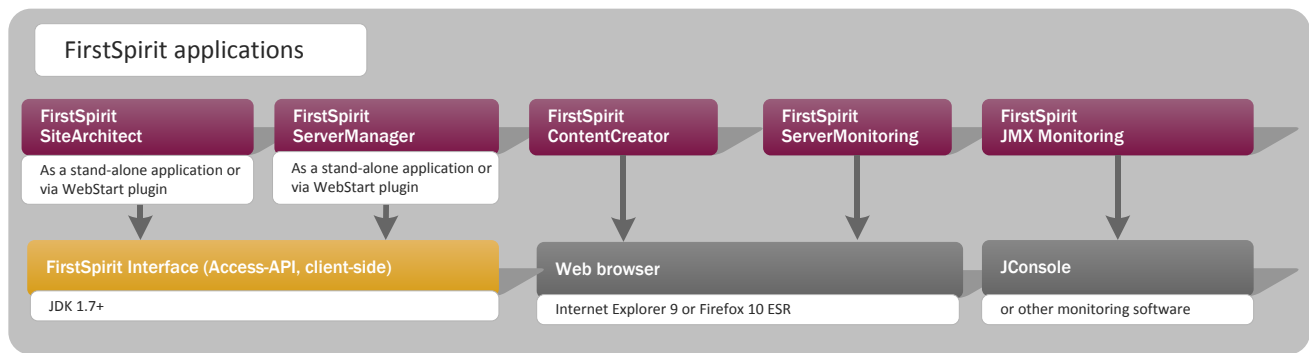


Figure 1-5: FirstSpirit editorial environments

The **documentation for administrators** describes all aspects of FirstSpirit V5.1 administration as well as the required administration environments and is, therefore, exclusively aimed at administrators. The FirstSpirit editorial environments (SiteArchitect and ContentCreator) are described in detail in separate documentations.

- **FirstSpirit ServerManager:** The FirstSpirit ServerManager 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 ServerManager can, e.g., be used to define users or integrate existing identity management systems, such as LDAP or Active Directory. Analogue to the SiteArchitect, the ServerManager is started and updated via Java Web Start (see chapter 7).
- **FirstSpirit ServerMonitoring:** The browser-based FirstSpirit ServerMonitoring is a web application for monitoring the FirstSpirit Server. Analogue to the ContentCreator, FirstSpirit ServerMonitoring is operated via a web browser (see chapter 8).



- **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 ServerMonitoring 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 ServerMonitoring, finer granular information is provided (see chapter 9).



This document is provided for information purposes only. e-Spirit may change the contents hereof without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. e-Spirit specifically disclaims any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. The technologies, functionality, services, and processes described herein are subject to change without notice.



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.

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 20 ff).

Chapter 3: This chapter describes the control of the FirstSpirit Server under Unix and Windows operating systems (page 21 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 28 ff).



Chapter 5: Configuration of the FirstSpirit web applications – especially project requirements, configuration and limitations of the ContentCreator – is described in this chapter (page 189 ff).

Chapter 6: The FirstSpirit start page is a web application for starting the FirstSpirit editorial and administration environments (page 194 ff).

Chapter 7: This chapter describes the project-wide functions which can be called via the ServerManager menu bar. The FirstSpirit ServerManager supports the general, administrative FirstSpirit tasks of the FirstSpirit administrator (page 211 ff).

Chapter 8: The browser-based FirstSpirit ServerMonitoring is used to monitor the FirstSpirit Server and displays current operating parameters (e.g. number of users, memory load) (page 370 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 449 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 494 ff).

Chapter 11: This chapter outlines the mechanisms for user permission assignment and permission check provided by FirstSpirit and their precise application (page 528 ff).

1.3 Supplementary documentation

1. FirstSpirit Technical Datasheet:
For detailed information about the system requirements of FirstSpirit Version 5.1 see FirstSpirit Technical Datasheet Version 5.1.
2. FirstSpirit Installation instructions:
For detailed information about installation and updating FirstSpirit servers please refer to "FirstSpirit Installation Instructions Version 5.1".
3. FirstSpirit Community:
Additional installation and administration instructions see FirstSpirit Community:
<https://community.e-spirit.com/community/developer?view=tags&tags=admindoc>



2 System Requirements and 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 applications are either managed via Java Web Start (SiteArchitect) or operated as a web application via a web browser (ContentCreator). 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, except a few files required for the system start.

For detailed information about the system requirements for FirstSpirit 5.1 please see current FirstSpirit Technical Datasheet, about installation and updating FirstSpirit servers please refer to "FirstSpirit Installation Instructions Version 5.1".



3 FirstSpirit Server control

To Control via the ServerMonitoring see Chapter 8.6.2.



Prior to starting and stopping the FirstSpirit Server, activate the maintenance mode (see Chapter 8.6.2.1 page 477).

3.1 Unix

3.1.1 GNU/Linux and Solaris 9

Start as `root`:

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

Stopp as `root`:

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

3.1.2 Solaris

Under Solaris FirstSpirit uses the Service Management Facility.

Start as `root`:

```
svcadm enable fs5
```

Stopp as `root`:

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

List processes as `root`:

```
svcs -p fs5
```



3.1.3 AIX

An entry in the file `/etc/inittab` (which occurs during installation with the identifier "fs5") can be used to start the FirstSpirit Server under AIX. Use the following calls to manually start and stop as root.

Start as root:

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

Stopp as root:

```
/opt/firstspirit5/bin/fs5.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 `fs5`, but deactivates the login for this user account. To activate possible login as `fs5` via SSH or Telnet, just enter the user account `fs5` password. Call as root:

```
passwd fs5
```

After logging in with the user account `fs5`, use the following calls to control the FirstSpirit Server:

Start as user `fs5`:

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

Stop as user `fs5`:

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

3.1.5 Generate a stack dump

Via the command:

```
fs5 dump
```

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

FirstSpirit ServerMonitoring provides an option for analysis. The thread dumps created via the "Threads" function can be analysed and displayed in a formatted view (see Chapter 8.6.5.5 page 488).



3.2 Windows

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

Start / All Programs / FirstSpirit 5.1:

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

3.2.1 Register / Deregister as system service via the start menu

Administrators can configure the FirstSpirit Server as a system service using the subitems "Register service" and "Deregister 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 Chapter 3.2.4 page 24).

The "Register service" function is required if the FirstSpirit Server is to be started or stopped via the service (see Chapter 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 the *FirstSpirit Installation Instructions* for further information on FirstSpirit uninstallation).



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

It is possible to start the FirstSpirit Server via the Windows start menu (menu "Server control").

Administrators can configure the FirstSpirit Server as a system service (see Chapter 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 Chapter 3.2.4 page 24).



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 Chapter 8.6.2.1 page 477).

3.2.4 Start /Stop in console mode

Click on the "Start Server in console" (menu "Server control") to open a console window in which the FirstSpirit Server is started.

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 Chapter 8.6.2.1 page 477).

3.2.5 Edit configuration files via the start menu

Certain FirstSpirit Server configuration files can be displayed and edited via the start menu, sub menu "Configuration".





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

Click on the desired entry

- Configure the Java Wrapper (Chapter 3.2.5.1 Page 25)
- Configure licence (Chapter 3.2.5.2 Page 25)
- Configure the server (Chapter 3.2.5.3 Page 26)

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 Chapter 4.3.1.1 page 33). The file should, if possible, be configured via FirstSpirit ServerMonitoring (see Chapter 8.6.1.5 page 473).

3.2.5.2 Configure licence

Click on the entry to open the configuration file `fs-license.conf`. See Chapter 4.3.5 (page 101) 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 ServerMonitoring (also see Chapter 8.6.1.2 page 471).



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.com>.

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 Chapter 4.3.1 (page 32) for a description of the parameters. The file should, if possible, be configured via FirstSpirit ServerMonitoring (see Chapter 8.6.1.1 page 471). Certain changes to the configuration file `fs-server.conf` require a server restart.

3.2.6 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 System¹. 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 Chapter 4.3.2.4 page 81 for the configuration).
- `fs-server.log`: Log file for the output messages of the FirstSpirit Server (see Chapter 4.3.6 page 103 for the configuration).

¹ <https://helpdesk.e-spirit.de/>



3.2.6.3 FirstSpirit start page

Click on the entry to open the FirstSpirit start page for starting the FirstSpirit applications (see chapter 6 Page 194).

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 26).

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 ServerMonitoring. ServerMonitoring additionally provides functions for analysing the stack dumps (see Chapter 8.6.5.5 page 488).



4 FirstSpirit Server configuration

4.1 File system organisation

In the following chapter, the directory structures of the server installation, the configuration file names and the web application structure will be explained.

4.1.1 File names

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



4.1.2 Web applications

- Start page and SiteArchitect (dynamic preview): fs5root
- Staging (local preview system, static preview): fs5staging
- ContentCreator and ContentCreator preview: fs5webedit
- ServerMonitoring: fs5webmon
- Preview (not ContentCreator): fs5preview

Generated project content is stored by FirstSpirit Server in the default generation directories (fs5staging, fs5preview and fs5webedit) or in local project generation directories. Access to these generation directories is protected, which means that when calling this content, user authentication is required as long as it has been configured (for information on configuring the login process, see Chapter 4.3.4 page 85).

These security measures apply to all global and local project staging web applications.

4.1.3 Directory structures

- System areas which are completely overwritten during server update:
 - ~fs5\bin Start environment incl. required system binaries
 - ~fs5\server Java environment incl. all libraries
 - ~fs5\web FirstSpirit web applications
- Configuration areas which can be changed by the user for the system configuration:
 - ~fs5\conf FirstSpirit configuration files (see Chapter 4.3)
- Data areas:
 - ~fs5\data All project data incl. modules, etc.
- Temporary file area & exports:
 - ~fs5\archive Archived files (see Chapter 7.5.10.1 page 402)
 - ~fs5\backup Project exports for data backup
 - ~fs5\export Project exports for interactive export/import processes
 - ~fs5\log Log files
 - ~fs5\work Temporary FirstSpirit files
 - ~fs5\web\fs5staging\projects Staging with web apps
 - ~fs5\web\fs5preview\preview_cache Preview cache
 - ~fs5\web\fs5webedit\preview_cache Preview cache ContentCreator



4.2 General configuration information

FirstSpirit is configured via configuration files located in the installation directory of the FirstSpirit Server (see Chapter 4.3 page 32).

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 ServerManager:** The FirstSpirit ServerManager facilitates editing of the configuration settings for database connection configuration (fs-database.conf) and login configuration (fs-jaas.conf) (see Chapter 6 page 194). All changes to these two configuration files carried out via the ServerManager are automatically stored in the respective configuration file and updated on the server.
- 2) **Via FirstSpirit ServerMonitoring:** Further configuration settings can be carried out via FirstSpirit ServerMonitoring or the JMX Console (see chapter 8 page 449 and chapter 9 page 494). Analogue to the ServerManager, 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 Chapters for further details). A server restart might be necessary to implement the changes (see Chapter 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 Chapter 9.4 page 501 for an example).

The following Chapters refer to the respective configuration possibility within the ServerManager, the JMX Console or ServerMonitoring.

The configuration examples in the following Chapters 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 Chapter 4.3.1.1 page 33).





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.



In general, additions should only be made to the parameters in the actual configuration files that are to be modified. Simply copying configuration examples from the manuals can result in the default values being overwritten, which is not always desirable



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 (Chapter 4.3.1).
- `fs-wrapper.conf` FirstSpirit Server start configuration (Chapter 4.3.1.1)
- `fs-database.conf` Database connection configuration (Chapter 4.3.3)
- `fs-jaas.conf` Authentication configuration (Chapter 4.3.4)
- `fs-license.conf` FirstSpirit licence configuration (Chapter 4.3.5)
- `fs-logging.conf` Logging configuration (Chapter 4.3.6)
- `fs-webapp.xml` Configuration settings Jetty (Chapter 4.3.7)
- `fs-update.conf` Necessary for the server update (Chapter 8.6.2.3); generally, this file need not to be modified and it must not be deleted.

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 ServerMonitoring (see Chapter 8.6.1.1 page 471). 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 ServerMonitoring) 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 (Chapter 4.3.1.1 page 33)
- Server (Chapter 4.3.1.2 page 37)
- ServerMonitoring (Chapter 4.3.1.3 page 41)
- Thread Pool (Chapter 4.3.1.4 page 41)
- Thread Queues (Chapter 4.3.1.5 page 43)
- JAAS (Chapter 4.3.1.6 page 45)
- Web Applications (Chapter 4.3.1.7 page 46)
- Preview (Chapter 4.3.1.8 page 49)
- Mail (Chapter 4.3.1.9 page 51)
- LDAP (Chapter 4.3.1.10 page 52)
- Storage Engine Properties (Chapter 4.3.1.11 page 56)
- CacheManager (Chapter 4.3.1.12 page 60)
- Internal Database (Chapter 4.3.1.13 page 60)
- ContentCreator configuration (Chapter 4.3.1.14 page 61)
- Miscellaneous (Chapter 4.3.1.15 page 62)
- Webstart configuration (Chapter 4.3.1.16 page 64)
- JumpToServlet (Chapter 4.3.1.17 page 66)
- JMX (Chapter 4.3.1.18 page 67)
- Authentication cookies (Chapter 4.3.1.19 page 69)

4.3.1.1 Area: Communication

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

HOST (optional): if an external application server is used instead of the web server integrated in FirstSpirit, the host name or IP address of the FirstSpirit Server is entered here. Servlets on the external application server connect to the `SOCKET_PORT` of the FirstSpirit Server at this address. This parameter is entered automatically in the `web.xml` file of the FirstSpirit servlet during startup.



HTTP_PORT: HTTP port of the FirstSpirit server (required for standard communication between the FirstSpirit client and FirstSpirit server).

URL (optional): the URL specified here for the FirstSpirit server start page is used at different places, for example

- in automatically sent e-mail messages containing a URL when the client is started. These types of e-mail messages are sent within defined workflows, such as in the case of status changes.
- when using the "Copy FirstSpirit address" function in SiteArchitect ("Extras" menu).
- for determining a reference within a FS_BUTTON input component.
- when switching from one web application to another (for example from fs5root in fs5webmon)

Usually, the URL is determined automatically, but this will not work if the server is recognized by many host names.

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

The parameter `URL` always needs to be set in case of one of the above mentioned applications (for example workflow e-mails) and an external web application server (e.g. Tomcat) are to be used.

The following parameters also need to be set in this case. The values are appended to the automatically determined URL or to the URL defined by the `URL` parameter and taken into account when a connection is being established.

`fs.url.hostname` (optional): this parameter is used to specify a host name for the client connection.

`fs.url.socketport` (optional): this parameter is used to specify the port when the client connection is to be established in socket mode (`%FIRSTspiritSOCKETURL%` placeholder in workflow e-mails).

`fs.url.httpport` (optional): this parameter is used to specify the port when the client connection is to be established in HTTP mode (`%FIRSTspiritURL%` in workflow e-mails).



`fs.url.usehttps` (optional): if the HTTPS URL is to be taken into account during a call, this parameter must be set to `true`. The default value is `false`

Example:

From the following configuration in `fs-server.conf`

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

the placeholder `%FIRSTspiritURL%` would become:

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

the placeholder `%FIRSTspiritSOCKETURL%`:

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

These parameters usually do not have any impact when used e.g. via the function "Switch to FirstSpirit address" in SiteArchitect in a project that has already been started.



*If the connection settings are activated on the start page (see Chapter 6.3.3.1 page 200), according to the order of evaluation (item **Evaluation order**) described in Chapter 7.3.8 page 261, the `fs.url` parameter is not taken into account. Unlike this order of evaluation, even though the corresponding parameters defined in the server properties in the "Web Start" and "Start Page" areas are ignored when the connection settings are disabled, the `fs.url` parameter is not.*

SOCKET_PORT: TCP port where the FirstSpirit server waits for connections for FirstSpirit's own socket protocol. Used for internal communication between servlets and the FirstSpirit server and, if configured, for communication with FirstSpirit SiteArchitect. This parameter is also entered automatically in the `web.xml` file of the FirstSpirit servlet during startup.



`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 if, for instance, an external application server is to be used.

`SOCKET_HOST` (optional): host name for the `SOCKET_PORT` bind address in order to limit the server to one IP address when required. If no value is passed, the server binds to all of the host IP addresses.



The `SOCKET_HOST` parameter (= the interface to which the socket listener is to bind) can only be used when the `HOST` parameter (= host name over which the server can be reached externally) has also been correctly configured, i.e. mapped to the same server network interface.

`SYMBOLIC_HOSTNAME` (optional): symbolic host name of the FirstSpirit server. This host name is only used for display on the start page and serves no other purpose.

`ALLOWED_ENCRYPTIONS` (optional): this parameter can be set for the user to specify what type of encryption must be used in the user's connection settings (see Chapter 6.3.3.1 page 200). When using TLS, the value 1 is set; when using DH_ARC4, 2 is set. If no encryption is to be used, the value 0 must be set. Any combination of the parameters 0, 1 and 2 may be used. Example:

`ALLOWED_ENCRYPTIONS=1,2`

If the encryption set by the user does not match the encryption specified here, when logging onto SiteArchitect or when attempting to use the ServerManager, the user will receive an error message stating that the connection parameters were not set correctly; communication between the client and server will not be possible.



TLS encryption (parameter value 1) cannot be used for the `ALLOWED_ENCRYPTIONS` parameter if WebSphere is used as the application server for the fs5root web application. In this case, RC4 encryption (parameter value 2) is available for SOCKET or HTTP implementation or HTTPS use. When using HTTPS, encryption is not necessary and therefore parameter value 0 is to be set.



4.3.1.1.1 Note on internal communication (IP multicast)

The FirstSpirit Server sends and receives IP multicast messages to determine which servers are accessible within a network. The multicast messages are sent and received under the address "239.192.34.16" for IPv4 and under the address "ff15::efc0:2210" for IPv6 (each with the TCP port 23416).

4.3.1.2 Area: Server

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

backup_files: number of backup versions saved to the internal structure files. This affects only the store and structure data and not the content data. All content data are under separate version control. Increasing the value improves the possibility of error correction, but also results in a higher amount of disk space allocated on the computer (default value is 50).

cyclicSaveTime: specifies the time in seconds after saving changes to the history, statistics, tasks or user settings, for instance. Cyclical saving only takes place when the relevant data has been modified (default value is 60 sec.).

JNLP_SERVLET_URL: the value defined here is required for generating FirstSpirit URLs as links for starting the client within e-mail messages. Example:

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

JNLP_REDIRECT: this parameter can be set to 1 if there is a problem with authentication via Kerberos (see 4.3.4.5 page 90) in conjunction with Java Web Start (JNLP) and Microsoft Internet Explorer. (Default value is 0.)

DTO_LRU_SIZE: this parameter defines the size of the DTO cache for a project on the server. Here the last project tree objects used are preserved. The value defines the number of objects managed in the cache (default value is 512 store elements).



CLIENTAPP_PATH: this parameter is required only for the rollout of native client applications (see Chapter 4.10.1 page 184). Default value: `FirstSpirit5\data\clientapp`.

CLIENT_HOME_DIR: This specifies the path to the directory in the workstation file system that is to be used for storing the client applications (see Chapter 4.10.2, page 185). Absolute (e.g. `CLIENT_HOME_DIR=C:/test`) or relative paths can be used. For relative paths, the character `~` can be used as a placeholder at the beginning of the path, e.g. `~/myclientapps`. `~` is then replaced by the current user home directory that is specific to the operating system.

If the parameter is not specified, client applications are rolled out to the user home directory specific to the operating system by default (e.g. `c:\users\name\.firstspirit_5.1R4`). Each version of FirstSpirit has its own directory. The directory name contains the particular FirstSpirit major, minor, and release version numbers, e.g. `\.firstspirit_5.1R4`.

Default value:

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



These directories are not deleted by the system when the version is changed and need to be removed manually, if required (e.g. to free up disk space).

Integrated preview (Internet Explorer only): If Internet Explorer is used for the "Integrated preview" functionality, the client application executables are not rolled out to the directory referred to above. Instead, they are placed in a temporary directory under the user home directory, e.g.:

```
c:\users\name\AppData\Local\Temp\FirstSpirit_123456536456
```

This behavior (which is specific to Internet Explorer) cannot be prevented by FirstSpirit. If using Internet Explorer, you should ensure that the appropriate execution permissions have been configured for the temporary directory as well (see Chapter 4.10.5, page 188; for information on the roll-out process, see Chapter 4.10.2, page 185).

CLIENT_HOME_DIR_WINDOWS: path to the directory in the file system where the client applications are to be rolled out to Windows operating system workstations.

Example:

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

If only `CLIENT_HOME_DIR_WINDOWS` is specified, the client application is rolled out to the specify operating system user home directory.



`CLIENT_HOME_DIR_LINUX`: path to the directory in the file system where the client applications are to be rolled out to Linux operating system workstations (see parameter `CLIENT_HOME_DIR_WINDOWS`).

`CLIENT_HOME_DIR_MAC`: path to the directory in the file system where the client applications are to be rolled out to Macintosh operating system workstations (see parameter `CLIENT_HOME_DIR_WINDOWS`).

`CLIENT_HOME_DIR_AIX`: path to the directory in the file system where the client applications are to be rolled out to AIX operating system workstations (see parameter `CLIENT_HOME_DIR_WINDOWS`).

`CLIENT_HOME_DIR_SOLARIS`: path to the directory in the file system where the client applications are to be rolled out to Solaris operating system workstations (see parameter `CLIENT_HOME_DIR_WINDOWS`).

`workflow.task.cache`:

`workflow.model.cache`: the type defined here specifies how long a task or workflow model is to be retained in the cache. A distinction is made here between `WEAK` and `SOFT`. If `WEAK` is specified, the objects are deleted directly from the cache as soon as they become obsolete. When `SOFT` is specified, the objects are retained in the cache, regardless of the VM used, as long as there is sufficient disk space. (If there is a large amount of disk space, it is usually advantageous to use `WEAK` as the type.) The LRU size is appended to the type in KB, separated with an underscore character. (Default values: `workflow.task.cache=SOFT_1024` and `workflow.model.cache=SOFT_128`).

`indexing.maxNoOfAssociations`: The value specified here limits the recursion depth when indexing datasets (that refer to each other in sequence) (default value 1024). Indexing is canceled when the defined value is reached. The data that has not been indexed at this point is not transferred to the index.

Note: In principle, indexing datasets in FirstSpirit is not problematic. Large recursion depths and a lengthy associated indexing process occur, for example, if there are a lot of cross-references between the datasets and these are made visible in an input component (`FS_LIST`, `FS_DATASET`). In a "to n" link, this configuration leads to very extensive data structures – and this is then reflected in the length of the indexing process. Instead of generally limiting the recursion depth, the forms responsible for this in the project should be adapted instead.



`webserver.conf-migration`: this parameter is set automatically. It is managed by the FirstSpirit server and must not be modified or deleted.

`externalServerAdminGroup`: Use this parameter to assign server administrator permissions to one or more external groups (e.g. from LDAP). To do this, specify the respective group name. All members of this external group will then receive server administrator permissions in FirstSpirit.

In order to create more than one external server administrator group, a unique extension is attached to the "externalServerAdminGroup" key, e.g.

```
externalServerAdminGroup.1=  
externalServerAdminGroup.2=
```

Example for the LDAP definition of two external server administrator groups:

```
externalServerAdminGroup.1=CN=fs-crew,OU=FIRSTspirit,OU=Projekte,DC=e-  
spirit,DC=de  
externalServerAdminGroup.2=CN=fs-dev,OU=FIRSTspirit,OU=Projekte,DC=e-  
spirit,DC=de
```

This configuration overwrites configurations that may have been set in ServerManager for the relevant users (see Chapter 7.2.4 page 235).

The "server administrator" property is set for external users and group members every time they log in. For more information please see also

- Chapter 7.1 page 213
(Info about the different types of administrators)
- Chapter 7.2.4 page 235
(Server administrator permissions for internal users)
- Chapter 7.4.8.2 page 317
(Creating external groups)



4.3.1.3 Area: ServerMonitoring

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

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

AJAX_DATA_SYNC_TIMEOUT: defines the time interval in seconds in which the dynamically updated data (via Ajax) are regenerated within ServerMonitoring (default value is 20).

AJAX_IMAGE_RELOAD_TIMEOUT: defines the time interval in seconds in which the dynamically updated statistics (graphic) (via Ajax) are regenerated within ServerMonitoring (default value is 60).

SYSTEM_MONITORING_MAIL_RECIPIENTS: e-mail addresses of users who are to be notified of the FirstSpirit server system status. The separator used between addresses is ";".

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.getRuntime().availableProcessors()")
ThreadPool.minSize=
# maximum number of concurrent threads, if left empty the value is # set to
(#cores * 8)
ThreadPool.maxSize=
```

FirstSpirit Server is scaled automatically to the number of available cores after installation or after a change in hardware. The number of available processors is determined using

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

ThreadPool.minSize: defines the minimum size of the limited thread pool (see Figure 9-6 page 506).



`ThreadPool.maxSize`: defines the maximum size of the limited thread pool and thus the maximum number of tasks that can be run concurrently (see Figure 9-6 page 506). The more processors a server has, the higher the value that can be configured for `ThreadPool.maxSize`.

The value of the threads actually executed may be higher, since this limitation does not apply to high priority tasks (unlimited thread pool).

If no explicit values are specified for the parameters

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

values that are dependent on the number of available processors are used automatically.



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    maximum numbers of running tasks.
# - queueCapacity queue capacity (-1 = unbounded, 0 = no queueing # allowed).
#####
ThreadQueue.LOW.maxRunning=2
ThreadQueue.LOW.queueCapacity=128

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

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

The `ExecutionManager`, which manages a multitude of different classified queues (see Chapter 9.8 page 505), is responsible for executing tasks. Some of the queues can limit the number of active tasks through the use of parameters. For queues classified as `BOUNDED`, the queue capacity can also be limited.

`ThreadQueue.LOW.maxRunning`: of the resource-intensive tasks within this queue, only a few should be run concurrently. The number of tasks that may be run at the same time can be configured using the `maxRunning` parameter (default value is 2 threads).

`ThreadQueue.DEFAULT.maxRunning`: the number of this queue's tasks that can be run concurrently can also be limited using the `maxRunning` parameter. Since these tasks are not as resource-intensive, the value can be higher than that defined in `ThreadQueue.LOW.maxRunning` (default value is 25 threads).

`ThreadQueue.DEFAULT.queueCapacity`: the capacity of a queue classified as `DEFAULT` can be limited.

`ThreadQueue.BOUNDED.maxRunning`: the queue classified as `BOUNDED` can be configured using two parameters. The parameter `maxRunning` can be used to limit the number of active tasks (default value is 25 threads).



`ThreadQueue.BOUNDED.queueCapacity`: In addition, the capacity of a queue classified as `BOUNDED` can be limited using the parameter `queueCapacity`. If the value configured under `queueCapacity` is attained, additional tasks from the server are rejected. This means that when the server is under a heavy load, only a certain number of tasks is placed in the queue (default value is 50 tasks) (of that number, those defined under `maxRunning` are processed directly). All other tasks are rejected for the time being and are transferred to an internal rejection strategy.

If no explicit values are specified for the parameters

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

values that are dependent on the number of available processors are used automatically.



The values of the parameters `ThreadQueue.LOW.maxRunning` and `ThreadQueue.LOW.queueCapacity` are not dependent on the number of 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 41).



4.3.1.6 Area: JAAS – Login 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
```

This area is used to define important login configuration settings for different FirstSpirit applications (e.g. SiteArchitect login or preview requirement). A number of different authentication modules are available for this purpose (e.g. SSO). In the JAAS² area, the different applications are mapped to the connection modules, which were configured in the fs-jaas.conf configuration file (see Chapter 4.3.4 page 85). A convenient option for configuring this area is available in ServerManager (see Chapter 7.3.10 page 264), and via ServerMonitoring (see Chapter 8.6.1.8 page 476).

² Java Authentication and Authorization Service

(For more information, see: <http://www.oracle.com/technetwork/java/javase/jaas/index.html>)



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_WEBEDIT5_URL=${WEBAPP_WEBEDIT5_NAME}
WEBAPP_WEBMON_URL=${WEBAPP_WEBMON_NAME}
WEBAPP_STAGING_URL=${WEBAPP_STAGING_NAME}
WEBAPP_PREVIEW_URL=${WEBAPP_PREVIEW_NAME}

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

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

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

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

GLOBAL_WEBAPPS: this parameter is set automatically and contains the IDs of the global web applications available on the server (see Chapter 7.3.13.1 page 279). The parameter is managed by the FirstSpirit server and must not be modified or deleted. Additional configuration settings for the global web applications are saved using the parameters WEBAPP_*_NAME, WEBAPP_*_CONTEXT_NAME, WEBAPP_*_SELECTED_WEBSERVER, WEBAPP_*_CONFIG_CHANGED and WEBAPP_*_ACTIVE_WEBSERVER



where * stands for the ID of the respective global web application. These parameters are also managed by FirstSpirit and must not be modified or deleted.

WEBAPP_LOG_LEVEL: this parameter is used to define the logging level for web applications. The following values can be specified for this purpose: TRACE, DEBUG, INFO, WARN, ERROR. Only entries of the same or higher level as that specified are written to the log file. (See also 4.6.2 page 141; default value: WARN.)

WEBAPP_ROOT_URL: URL to the Start Page web application (fs5root). The value stored here is required for mapping within the internal servlet engine (see also 4.3.7.2 page 108). The value must always start with "/", followed by the symbolic name of the particular web application. The URL is required for links to different FirstSpirit applications. (Default value: /).

The following entry is required for operation under WebSphere:

`WEBAPP_ROOT_URL=/fs5root`

(see also 4.6.1 page 139.)

WEBAPP_WEBEDIT5_URL: URL for the ContentCreator (fs5webedit) web application within the start page. The value stored here is required for mapping within the internal servlet engine (see also Chapter 4.3.7.2 page 108). The value must always start with "/", followed by the symbolic name of the particular web application. The URL is required for links to different FirstSpirit applications (default value: `/${WEBAPP_WEBEDIT5_NAME}`).

WEBAPP_WEBMON_URL: URL for the "ServerMonitoring" (fs5webmon) web application within the start page. The value stored here is required for mapping within the internal servlet engine (see also Chapter 4.3.7.2 page 108). The value must always start with "/", followed by the symbolic name of the particular web application. The URL is required for links to different FirstSpirit applications (default value: `/${WEBAPP_WEBMON_NAME}`).

WEBAPP_STAGING_URL: URL for the FirstSpirit Staging (fs5staging) web application. The value stored here is required for mapping within the internal servlet engine (see also Chapter 4.3.7.2 page 108). The value must always start with "/", followed by the symbolic name of the particular web application. The URL is required for links to different FirstSpirit applications (default value: `/${WEBAPP_STAGING_NAME}`).

WEBAPP_PREVIEW_URL: URL for the FirstSpirit Preview (fs5preview) web application. The value stored here is required for mapping within the internal servlet engine (see also Chapter 4.3.7.2 page 108). The value must always start with "/", followed by the



symbolic name of the particular web application. The URL is required for links to different FirstSpirit applications (default value: `/${WEBAPP_PREVIEW_NAME}`).

WEB_DIR: path to the FirstSpirit web directory for use with all FirstSpirit web applications (fs5root, fs5preview, fs5staging, fs5webedit, fs5webmon) (for `${}` notation, see Chapter 4.2 page 30).

WEBAPP_ROOT_PATH: path to the directory in the file system where FirstSpirit Server stores the FirstSpirit Start Page (fs5root) web application (for `${}` notation, see Chapter 4.2 page 30).

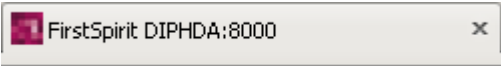
WEBAPP_WEBEDIT5_PATH: path to the directory in the file system where FirstSpirit Server stores the FirstSpirit ContentCreator (fs5webedit) web application (for `${}` notation, see Chapter 4.2 page 30).

WEBAPP_WEBMON_PATH: path to the directory in the file system where FirstSpirit Server stores the FirstSpirit ServerMonitoring (fs5webmon) web application (for `${}` notation, see Chapter 4.2 page 30).

WEBAPP_STAGING_PATH: path to the directory in the file system where FirstSpirit Server stores the FirstSpirit Staging (fs5staging) web application. FirstSpirit Server also stores all generated project files in this directory (for `${}` notation, see Chapter 4.2 page 30).

WEBAPP_PREVIEW_PATH: path to the directory in the file system where FirstSpirit Server stores the FirstSpirit Preview (fs5preview) web application. FirstSpirit Server also stores the generated preview pages in this directory.

WEBAPP_ROOT_TITLE: this parameter can be used to change the title of the FirstSpirit start page. The default display value is `FirstSpirit ${HOSTNAME}:${HTTP_PORT}`. The FirstSpirit Server start page in this example thus appears in the browser as follows:

follows: 





If the host name of the FirstSpirit server, defined symbolically using the parameter `SYMBOLIC_HOSTNAME`, is to be used (see Chapter 4.3.1.1 page 33) when `WEBAPP_ROOT_TITLE` is not specified, `WEBAPP_ROOT_TITLE=FirstSpirit ${SYMBOLIC_HOSTNAME}` can be set, for instance. Replacement is not automatic, since `SYMBOLIC_HOSTNAME` is an optional parameter that can also be empty.

`CLIENTJAR_URL`: URL of the FirstSpirit client Jar file.

4.3.1.8 Area: Preview

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

`preview.internalDelivery`: a comma-separated list of file extensions to be supplied directly by the servlet engine can be specified here. Files that are not in the list are supplied to the web server via an internal redirect. (This can be an Apache web server, for instance, which then handles processing of the file, e.g. PHP.) Default value: *. This is used to allow the servlet engine to supply all files with an extension that is not in the `preview.externalDelivery` parameter list.

`preview.externalDelivery`: this parameter is used to specify a comma-separated list of file extensions that are to be supplied by an external web server and **not** by the servlet engine. This parameter is only taken into account when the `preview.internalDelivery` parameter is set to *. By default, the parameter is empty and must be set as needed.

`preview.externalDeliveryURL`: this parameter is used to specify the URL to the external web server that will be used for file types that were not defined in the `preview.internalDelivery` parameter (e.g. PHP or ASP). The URL is composed of the server name and the port, which is configured in Chapter 4.5.1 under "Virtual web server" (page 116). Example:

`preview.externalDeliveryURL=http://fs5.yourdomain.net:80`. By default, the parameter is empty and must be set as needed.

`preview.cacheTimeout`: the generated pages that are stored temporarily in the `cacheDir` are only valid for the time interval defined here (in seconds). If the interval has lapsed,



the preview pages (when requested) are regenerated and cached again. All files that are older than the set time interval are removed from the cache. The default value is 1200 seconds; the minimum value is 60.

`preview.cacheFileWithTimestamp`: this parameter is used to add a time stamp to file names.

This is important, for instance, when using IBM WebSphere so that JSP files can be recompiled correctly when the content of the file has changed after being compiled previously. For this purpose, a comma-separated list of file extensions that are to have a time stamp appended to the file names is specified for `preview.cacheFileWithTimestamp`, e.g.

`preview.cacheFileWithTimestamp=jsp,jsf`

In order to add the time stamp to all file names, the parameter can be set to `*`. The parameter is empty by default. The parameter must be set when using IBM WebSphere.

`preview.enforceDelivery`: usually files that are still in the cache of a web browser and have not changed since they were last queried are not resent by the server. Instead, the server sends the HTTP status code 304 ("Not modified"). The `preview.enforceDelivery` parameter is used to specify file types for which a response using the status code 304 is never to be sent. This makes it possible to force a delivery for these file types. For the parameter, multiple file extensions can be specified by separating them with a comma. Default setting:

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



4.3.1.9 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. This is required, since the server sends an e-mail upon request reporting on the result after delivery or generation of a site.

`mail.smtp.port:` this parameter is used to specify the port that the SMTP server should use to send e-mail messages. Port 25 is used by default.

`mail.default-recipient:` the default e-mail address.

`mail.sender:` the e-mail address that will be used as the sender for all FirstSpirit Server e-mail. If no e-mail address is specified here, an e-mail address using the following format will be used:

```
cmsserver@[Hostname],  
where the host name is determined using the method  
InetAddress.getLocalHost().getHostName().
```

`mail.subject.charset:` this parameter is used to set the encoding for the subject line in e-mail messages sent from FirstSpirit. ISO-8859-1 is used by default.

`mail.mime.charset:` this parameter is used to set the encoding for the body in e-mail messages sent from FirstSpirit. The encoding of the operating system running on FirstSpirit Server is used by default.



4.3.1.10 Area: LDAP

Different LDAP configurations (known as sections) can be created in FirstSpirit Server. The name of a section is defined by the entry in the fs-jaas.conf configuration file (see Chapter 4.3.4 page 85). A section is configured in the fs-server.conf configuration file (LDAP area). The section name (from the fs-jaas.conf file) is specified before each configuration parameter. The section name must follow the "LDAP_n" format, where "n" is the section number that is numbered sequentially starting with 1. If only 1 LDAP section is used, the section can also be called "LDAP".

```
LDAP_n.parameter=wert
```

The names of the LDAP attributes are case sensitive and must be entered using the same case used in the LDAP directory.



To comment out the following parameters, an empty string must be specified as the value.

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

Commenting out this parameter using # does not work, since in this case the default value is used.

"LDAP" was selected as the section name in the following example configuration for connecting to the LDAP server of the Microsoft Active Directory (using LDAP.AUTHENTICATION=SEARCH_BIND). Depending on the configuration of the fs-jaas.conf file (see Chapter 4.3.4), other section names can also be selected, and many different LDAP sections can be defined at the same time.

```
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: description of the corresponding LDAP section, e.g. the domain name. The description appears in the "Edit user" dialog (see Chapter 7.2.4.2 page 237).

LDAP.HOST_URL: LDAP URL of the LDAP section in the format `ldap://hostname` (if `LDAP.SSL=false`) or `ldaps://hostname` (if `LDAP.SSL=true`). To improve fail-safe performance, you can register multiple LDAP servers that have to provide the same LDAP data.

LDAP.SSL: the encrypted SSL transfer can be enabled (value: `true`) or disabled (value: `false`) here (see 4.4.3).

LDAP.AUTHENTICATION: there are different server login options available. Possible values are:

- **BIND:** the name and password are sent to the LDAP server. The "Distinguished Name" (DN), i.e. the unique user identification key, must be known within the LDAP server. If the DN exists, the password passed is checked using the "Bind" operation. BIND can only be used when the LDAP DN's of the user accounts are all inside the same LDAP folder. The example for the `LDAP.BIND.DN` parameter (see below) contains the folder with the DN

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

If the user accounts are distributed across different folders, either `SEARCH_BIND` must be entered into the `fs-server.conf` file or a unique LDAP section must be entered into the `fs-server.conf` file for each of the user folders.

- **SEARCH_BIND:** if the "Distinguished Name" (DN) of a user is unknown, or the user accounts are entered in different branches of the LDAP tree, you can search for it within a subtree of the LDAP server. A search filter must be defined to do this. 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 "uid" attribute is the same as the login name entered. The start node is the node with the DN "ou=department,dc=mycompany,dc=com". The login is successful as soon as a matching user account is found on the `SEARCH.FILTER` and the user password passed is correct.



- `SEARCH_COMPARE`: the function of this option is equivalent to `SEARCH_BIND`; in this case, however, it is not the password attribute that is used for authentication, but rather any other LDAP attribute. Example:

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

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



When using `LDAP.AUTHENTICATION=SEARCH_BIND` or `LDAP.AUTHENTICATION=SEARCH_COMPARE`, it is usually necessary to provide entries for `LDAP.SEARCH.BIND_DN` and `LDAP.SEARCH.BIND_PASSWORD` (see below). Entry is not required if the LDAP server can be queried without authentication. However, this actually does not occur in production environments.

`LDAP.BIND.DN`: DN of the user accounts that can be used to log onto FirstSpirit. This parameter is only useful in conjunction with `LDAP.AUTHENTICATION=BIND`. `$USER_LOGIN$` is entered as a placeholder 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=Benutzer,ou=Dortmund,dc=e-spirit,dc=de
LDAP_1.IMPORT_USER=TRUE
LDAP_1.IMPORT_USER.LOGIN_ATTRIBUTE=sAMAccountName
LDAP_1.IMPORT_USER.NAME_ATTRIBUTE=displayName
LDAP_1.IMPORT_USER.EMAIL_ATTRIBUTE=mail
LDAP_1.IMPORT_USER.GROUP_ATTRIBUTE=memberof
LDAP_1.IMPORT_USER.PHONE_ATTRIBUTE=telephoneNumber
LDAP_1.IMPORT_USER.ABBREVIATION_ATTRIBUTE=initials
```

`LDAP.SEARCH.BIND_DN`: LDAP DN of a technical user account used to search the LDAP server in order to find a DN of a FirstSpirit user who is to be logged in.

`LDAP.SEARCH.BIND_PASSWORD`: password for 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`: this parameter is used to define a search filter. The filter:
`SEARCH.FILTER=(cn=$USER_LOGIN$)`
searches, for instance, for all entries in the LDAP tree in which the attribute "cn" is the same as the login name entered in FirstSpirit. The start node is the DN specified for `SEARCH.BASE_DN`.

`LDAP.IMPORT_USER`: in addition to strict authentication, it is possible to import any LDAP attributes into the user attributes of a CMS user. To do this, the value of the parameter `LDAP.IMPORT_USER` must be set to `TRUE`.

`LDAP.IMPORT_USER.LOGIN_ATTRIBUTE`: this assignment imports the login name of an LDAP user for a FirstSpirit user. The LDAP name assigned here is imported automatically after the particular user logs in for the first time. The Active Directory attribute `sAMAccountName` is usually not unique across all domains if multiple LDAP sections (`LDAP_1`, `LDAP_2`, ...) are present, i.e. users may be authenticated against multiple user domains. The complete attribute `userPrincipalName`, which includes account as well as domain names, should be used here. If an invalid value is specified for this parameter (e.g. LDAP attribute returns an "void" value, invalid LDAP attribute, etc.), this is logged in the server log file as follows:

```
INFO 17.05.2010 14:50:24.102
(de.espirit.firstspirit.server.usermanagement.LDAPAuthentication): [LDAP]
ignoring empty LOGIN_ATTRIBUTE value!
```

`LDAP.IMPORT_USER.LOGIN_ATTRIBUTE`: this assignment imports the user name of an LDAP user for a CMS user. The LDAP name assigned here is imported automatically after the particular user logs in for the first time.

`LDAP.IMPORT_USER.EMAIL_ATTRIBUTE`: this assignment imports the e-mail address of an LDAP user for a CMS user. The LDAP e-mail address assigned here is imported automatically when the particular user logs in for the first time.

`LDAP.IMPORT_USER.GROUP_ATTRIBUTE`: the LDAP group attribute allows a user to assign a particular FirstSpirit group of a FirstSpirit project automatically using his or her group membership in LDAP.

The name of the LDAP attribute is specified which contains the LDAP DN of the LDAP groups in which the particular user is a member. The attribute is read out again each time the user logs in so that the group membership can be imported into FirstSpirit. All externally marked FirstSpirit groups are assigned to the user account and have an "external name" assigned in FirstSpirit that matches the LDAP DN of the LDAP group. Instead of an LDAP DN, any string can be used;



however, the LDAP server typically maps group membership via DN's.

`LDAP.IMPORT_USER.PHONE_ATTRIBUTE`: this assignment links the phone number of an LDAP user to a CMS user. The LDAP phone number assigned here is imported automatically the first time the particular user logs in.

`LDAP.IMPORT_USER.ABBREVIATION_ATTRIBUTE`: this assignment links the abbreviated name of an LDAP user to a CMS user. The LDAP abbreviation assigned here is imported automatically the first time the particular user logs in.

For additional information on connecting an LDAP server, see Chapter 4.4 page 110.

4.3.1.11 Area: Storage Engine Properties

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

# Repository encryption
#-----
# Enable repository encryption. A keyfile must be set to enable encryption.
repository.encryption=0

# Path to master key file.
# The content of the file is read with UTF-8 encoding. Leading and trailing
whitespace is ignored. The key characters
# are processed with the PBKDF2WithHmacSHA1 password-based key derivation
function.
repository.encryption.keyFilePath=

# The symmetric encryption algorithm name, mode and padding.
#
http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#Cipher
#
http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html#SunJCEProvider
# repository.encryption.algorithm=AES/CBC/PKCS5Padding
repository.encryption.algorithm=AES/CTR/NoPadding

# Keysize to use for the specified encryption algorithm. See SunJCE provider
documentation for allowed keysizes.
#
http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html#SunJCEProvider
```



```
#
http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.
html#importlimits
repository.encryption.keySize=128
```

All Berkeley DB properties can also be used in the `fs-server.conf` configuration file. To do this, the prefix "repository." must always be specified before the particular property.

The relevant properties can be found in the Berkley DB documentation³.

`repository.sharedCache`: The value of this parameter is set to 1 by default. This allows the Berkley DBs of all projects to share one cache on a server. In particular, this makes it easier to manage the Berkley cache on servers with several projects and is more effective overall. If each project is to have its own cache for the Berkley DB, the value can be set to 0.

Configuring the file repository so it is encrypted (FirstSpirit Version 5.1R4 and higher):

`repository.encryption`: In the FirstSpirit repository, content, structures, and media from FirstSpirit projects are usually saved unencrypted (default value: 0). To configure encrypted storage of this data in the repository, you must set the parameter to a value of 1. The setting made here is used as a default setting for all new or imported projects:

- If it is set to 1, the "Encryption active" project setting (see Chapter 7.4.22 page 366) is enabled for all new or imported projects.
- If it is set to 0, the "Encryption active" setting (see Chapter 7.4.22 page 366) is disabled for all new or imported projects.

Pre-existing projects are not affected if the parameter is changed, that is, encryption must be enabled (or disabled) separately for existing projects by going to the project settings (see Chapter 7.4.22 page 366).

Before encryption can be enabled, you must create a global server key (see Chapter 4.8.2.1, page 153) and use the `repository.encryption.keyFilePath` parameter to configure the path to the server key file (see below).

`repository.encryption.keyFilePath`: If repository encryption is enabled (`repository.encryption=1`), this parameter is used to specify the path to the global server key, e.g.:

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



```
repository.encryption.keyFilePath=${cmsroot}/conf/fs5key.txt
```

The global server key must be at least eight bytes long. The content of the specified file must be encoded in UTF-8. White spaces at the beginning and end of the file are ignored. The path can be specified as absolute or relative (to the root directory of the FirstSpirit server). `${cmsroot}` can be used as a placeholder for the FirstSpirit root directory.

There is only one key file for each FirstSpirit server. In a cluster scenario involving master and slave servers, all the servers concerned must use the same key file.



Access to the global server key file should be properly secured to prevent unauthorized persons from accessing the repository contents. At the same time, this means that if the key file is damaged or lost, it will no longer be possible to access the contents of the repository.

`repository.encryption.algorithm`: The symmetric algorithm configured here is used to encrypt the contents of the project repository. Rather than making direct use of the server key file, this involves generating an internal key for the project repository. The process of generating the internal project key relies on the `PBKDF2WithHmacSHA1` key derivation algorithm (65536 iterations, 256-bit random salt, 256-bit key size⁴).

The preconfigured example value for encrypting the project repository is `AES/CTR/NoPadding` (this value is intended purely as a configuration example and is not to be construed as a recommendation). The actual encryption process is handled by the Java Cryptography Extension. Therefore, it is possible to utilize all the symmetric encryptions and modes that are supported by the Java platform used. For details of which algorithms, modes, and key sizes are possible, please see the JCE documentation:

- <http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#Cipher>
- <http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html#SunJCEProvider>

The value configured here is transferred to the project properties under "Repository", "Encryption algorithm" in ServerManager (for new or imported projects). For existing projects, the value can be configured within the project properties (see Chapter 7.4.22, page 366).

⁴ Password-based key derivation function 2



`repository.encryption.keySize`: This parameter can be used to configure the length of the key for encrypting the repository contents. The values configured here must be compatible with the configured algorithm (encryption, modes) and are dependent on the Java version used (see `repository.encryption.algorithm`). Once again, the JCE documentation (see above) provides details of which values are possible.

<http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html#SunJCEProvider>

The value configured here is transferred to the project properties under "Repository", "Encryption key size" in ServerManager (for new or imported projects). For existing projects, the value can be configured within the project properties (see Chapter 7.4.22, page 366).

Larger key sizes and stronger algorithms can be configured if "Unlimited Strength Jurisdiction Policy Files" are installed on the server:

<http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html#importlimits>

Configuring default values: As soon as a key file has been created and configured, all the encryption parameters (for each individual project) can be enabled and configured in the "Repository" area of ServerManager as well (see Chapter 7.4.22, page 366). All the values that are configured globally using `fs-server.conf` are defined as default values in the project configuration. This means that the globally configured values will be applied as default values for any projects that are created or imported as of this point. The global configuration does not affect existing projects, which must be configured via the project settings instead.

The parameters can also be tested within the project configuration.

For more information on repository encryption, see Chapter 4.8.2, page 153.



If changes are made to the `fs-server.conf` file, the server will need to be restarted (see Chapter 4.2 page 30).

4.3.1.12 Area: CacheManager

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

In this area, either the absolute or the relative (percentage) size of the cache required can be set. The value should not be set to 0.

`CACHE_SIZE`: absolute size of the cache in bytes. Valid values include 456458345, 128m or 4096k, for instance.

`CACHE_PERCENT`: size of cache as a percentage of the `-Xmx` value. Valid values include 30 or 0.5, for instance. The value is set to 25 by default.

4.3.1.13 Area: internal database

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

`internalDB.port`: TCP port where 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 bound. The default value is "" in order to ensure that all addresses are used.

FirstSpirit Server already includes a simple relational database system (Apache Derby), which is available immediately after the server is installed. However, this database is not suited for production mode and therefore should only be used for testing.

This bind address allows external clients or web applications from external servers to access the internal database (Derby). If the value for `internalDB.port` is set to 0, this function is disabled. This is the default configuration in order to ward off any potential security vulnerabilities.



The database access files are located in the fs-database.conf file (see Chapter 4.3.3 page 84).

An example for configuring the Derby database for use with external processors (e.g. web application with the FirstSpirit DynamicDatabaseAccess module in the external application server) can be found in Chapter 4.9.7.6 page 182.

4.3.1.14 Area: WEBedit configuration

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

`webedit.sessionCache.lru_size`: this parameter defines the size of the LRU⁵ cache for each individual ContentCreator session. The LRU cache is the cache for all FirstSpirit objects referenced within the project. The value defined here provides information on how long the referenced objects will remain in the cache. A distinction is made between "weak references" (value < 0), which are immediately deleted from the cache as soon as they are no longer referenced, and "soft references" (value >=0), which remain in the cache depending on the VM used as long as there is sufficient disk space available. (If there is a large amount of disk space, it is usually advantageous to use weak references.)

`webedit.globalCache.lru_size`: this parameter defines the size of the DTO⁶ cache of all ContentCreator sessions. The value defined here determines the number of store elements that can be stored in the DTO cache. The DTO cache contains strictly the data objects of a project. In this case, all users work on the same data objects (as opposed to the project objects within the stores, which are only valid locally within a session). The cache is therefore shared across all ContentCreator sessions (global cache).

`web.sessionCache.lru_size`: this parameter defines the size of the cache for web session project objects that are required in order to generate the preview. The value

⁵ LRU: Least Recently Used

⁶ DTO: Data Transfer Object



defined here determines the number of store elements that can be stored in the cache.

`webedit.maxConcurrentThreadsCount`: This parameter limits the maximum number of possible threads that can be executed in parallel on the application server (e.g. Tomcat) (default value: 128). Reasoning: Certain actions (e.g. a search request in ContentCreator) generate threads on the application server in the background. By default, the number of parallel threads is limited to 128. In particular individual cases, it may be advisable to increase this value.

4.3.1.15 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
```

BATCHPATH: this parameter defines the path to the script files that can be run during deployment, for instance.

LICENSE_EXPIRATION_WARNING_DAYS: this parameter defines the number of days before a FirstSpirit license expires that a daily license expiration warning will be issued. If the specified value is "0", no e-mail will be sent (default value is 30 days).

LICENSE_EXPIRATION_MAIL_ADDRESS: this parameter defines the e-mail address to which the license expiration warning message will be sent. If no e-mail address is defined, the e-mail will be sent to the administrator (see the `ADMIN_MAIL_ADDRESS` parameter in Chapter 4.3.1.9 page 51).

BACKUP_PATH: the path to the backup directory — the directory in the file system where FirstSpirit Server stores backups. Backups are created via the "Execute project backup" schedule entry (see Chapter 7.5.10.3 page 412) or the license-dependent



"FirstSpirit EnterpriseBackup" functionality (see the corresponding module documentation). Backups are stored in the FirstSpirit Server `backup` subdirectory by default.

`mail.backup-recipient`: this parameter allows for an e-mail address to be specified that is used when the parameters `backup.size.limit`, `backup.min.age` and/or `backup.max.age` (see below) are enabled.

`backup.size.limit`: this parameter can be set so that an e-mail is sent to the e-mail address defined by the `mail.backup-recipient` parameter if the size of the backup files exceeds the size specified. Examples of values include `500m` or `3g`. No value is specified by default, which means that the function is disabled.

`backup.min.age`: this parameter can be set so that backup files are deleted automatically after the specified period (in days) if the value defined by the `backup.size.limit` parameter is reached. No value is specified by default, which means that the function is disabled. To enable this function, a value must be specified. When backup files are deleted, an e-mail is sent to the e-mail address defined by the `mail.backup-recipient` parameter.

`backup.max.age`: this parameter can be set so that backup files are deleted automatically after the specified period (in days). No value is specified by default, which means that the function is disabled. To enable this function, a value must be specified. When backup files are deleted, an e-mail is sent to the e-mail address defined by the `mail.backup-recipient` parameter.

`hdd.directories`: this parameter specifies the directories to be monitored. The default value is the FirstSpirit root directory, which is `hdd.directories=${cmsroot}`. This is used when the value is not entered manually. See also 7.3.1 page 243, option "Directories for disk space check (comma separated)", etc.

`hdd.limit`: this parameter is used to specify the percentage at which point a warning e-mail message is sent to the server administrator. Values from `1` to `99` can be specified. If no warning e-mail is to be sent, the value can be set to `-1`

The default value is 90%, i.e. `hdd.limit=90`. This is used if the value is not specified manually or if the value specified is not between `1` and `99` or is `-1`.

`hdd.limit.active`: this value is used to prevent warning e-mail messages from being sent. In this case, `hdd.limit.active=false` must be set. The sending of messages,



however, is enabled by default when a percentage is specified.

`hdd.shutdown`: this parameter is used to specify the percentage at which point a warning e-mail message is sent to the server administrator and the server is shut down. Values from 1 to 99 can be specified. If the server should not be shut down, the value can be set to -1. If the selected value is equal to or less than `hdd.limit`, `hdd.shutdown` is set by the system to be 5% higher than `hdd.limit`. The default value is 95%, i.e. `hdd.shutdown=95`. This is used if the value is not specified manually or if the value specified is not between 1 and 99 or is -1.

`hdd.shutdown.active`: this value is used to prevent shutdown of the FirstSpirit Server. To prevent shutdown, `hdd.shutdown.active=false` must be set. Shutdown, however, is enabled by default when a percentage is specified.

`SERVICES`: this parameter is used to define the system services. (These services can be started when the server is started.)

4.3.1.16 Area: Web Start configuration

To start the FirstSpirit ServerManager and the FirstSpirit SiteArchitect, the Oracle Java Runtime Environment (JRE) is required, which contains Java Web Start. (JRE is usually installed automatically when JDK is installed. For information on the supported versions, see the *FirstSpirit Technical Data Sheet*.) The configuration is set using JNLP files:

- global (server-wide) via the FirstSpirit ServerManager (see Chapter 7.3.7 page 260) or via the parameters in the `fs-server.conf` configuration file (see below)
- user-specific via the connection settings on the FirstSpirit start page (see Chapter 6.3.3.1 page 200).

Web Start configuration for starting the FirstSpirit SiteArchitect: the parameters configured here affect all SiteArchitect type quick-start entries on the start page **JAVA** if no other parameters were explicitly defined for the entry in the "start page" area (see Chapter 7.3.8 page 261).

```
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 Web Start configuration (see Chapter 6.3.3.1 page 200).



TLS encryption (parameter value 1) cannot be used for the `webstart.client.encryption` parameter if WebSphere is used as the application server for the `fs5root` web application. In this case, RC4 encryption (parameter value 2) is available for SOCKET or HTTP implementation or HTTPS use. When using HTTPS, encryption is not necessary and therefore parameter value is to be set to 0.

ServerManager tab: configuration for the ServerManager 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 Web Start configuration (see Chapter 6.3.3.1 page 200).



TLS encryption (parameter value 1) cannot be used for the `webstart.admin.encryption` parameter if WebSphere is used as the application server for the `fs5root` web application. In this case, RC4 encryption (parameter value 2) is available for SOCKET or HTTP implementation or HTTPS use. When using HTTPS, encryption is not necessary and therefore parameter value is to be set to 0.



4.3.1.17 Area: JumpToServlet and ContentCreator ForwardAction

`allowedRedirectHosts`: at some points in FirstSpirit, redirect URLs are generated (e.g. links from remote projects in the preview or for the ContentCreator preview) that can potentially also refer to external URLs; for example:

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

or

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

URLs can be defined using the optional parameter `allowedRedirectHosts` to which a redirect is to be allowed. The following modes are possible:

```
allowedRedirectHosts=ALLOW_ALL
```

Redirects to all URLs are allowed without limitations.

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

This allows for the creation of a white list of allowed targets. The allowed URLs are specified as comma-separated:

```
allowedRedirectHosts=FS_SERVER
```

A white list of allowed URLs is created from the following sources:

1) `fs-server.conf`, parameter

URL	(Chapter 4.3.1.1 page 33)
JNLP_SERVLET_URL	(Chapter 4.3.1.2 page 37)
WEBAPP_ROOT_URL	(Chapter 4.3.1.7 page 46)
WEBAPP_WEBEDIT5_URL	(Chapter 4.3.1.7 page 46)
WEBAPP_WEBMON_URL	(Chapter 4.3.1.7 page 46)
WEBAPP_STAGING_URL	(Chapter 4.3.1.7 page 46)
WEBAPP_PREVIEW_URL	(Chapter 4.3.1.7 page 46)
preview.externalDeliveryURL	(Chapter 4.3.1.8 page 49)

2) in the web server configured server properties (ServerManager / "Server" / "Properties" / "Web Server", see Chapter 7.3.12 page 271).

This is the default setting.

If an attempt is made to call a URL that is not allowed, the HTTP status code 403 (Forbidden) is output.



4.3.1.18 Area: JMX

```
#####  
# JMX  
#####  
# Listen host and port. If port is empty, JMX is disabled  
jmx.host=${HOST}  
jmx.port=  
  
# SSL and keystore  
jmx.ssl=false  
jmx.ssl.needClientAuth=false  
javax.net.ssl.keyStore=  
javax.net.ssl.keyStorePassword=  
  
# User authentication and access level. If password file is empty, authentication is disabled  
jmx.password.file=  
jmx.access.file=
```

Configuration of the JMX connector for (remote) monitoring of the FirstSpirit Server JVM (see Chapter 9 page 494). JMX is used to query the system status and provides current FirstSpirit Server and Java system information. For example, `jconsole` (contained in JDK) or other system monitors that support the JMX protocol can be used as a client.

jmx.host: This parameter instructs the FirstSpirit Server to accept incoming requests at the specified `jmx.port` (see below) in accordance with the host name (or IP address) defined here. By default, it accepts all IP interface connections. `jmx.host` has to be configured if the JMX connector is to be restricted to a specific interface for security reasons. The information is also required if no remote JMX connection can be established from `jconsole`, for example, as the dedicated host name of the server cannot be resolved via DNS or has only been entered in `/etc/hosts` using a local IP address. In this case, the dedicated IP address or dedicated host name of the server must be entered here.



`jmx.port`: a free port number for the JMX / RMI connection (JMX connector). If no port is specified, access to JMX is not possible.

`jmx.ssl`: use of Secure Sockets Layer (SSL) is disabled by default (default value: `false`).

`jmx.ssl.needClientAuth`: use of client SSL authentication is disabled by default (default value: `false`). To enable client SSL authentication for remote monitoring via JMX, this parameter must be set to `true`.

`javax.net.ssl.keyStore`: this parameter is used to specify the file system path to the keystore.

`javax.net.ssl.keyStorePassword`: password for the keystore file.

`jmx.password.file`: enables user authentication for JMX access to the FirstSpirit server. This parameter is used to specify the file system path to the JMX password file. The JMX password file manages different roles/users and their passwords. Note: Since the passwords in this file are saved as plain text, the default authentication information should not be stored here; roles and passwords defined specifically for JMX access should be stored here instead (see `jmx.access.file`). The JRE contains a template for a password file called `jmxremote.password.template`. This template can be copied to `JRE_HOME/lib/Management/jmxremote.password` and can be expanded to include passwords for the roles that are defined in the JMX access file. If no value is specified (default state), JMX authentication is disabled.

`jmx.access.file`: this parameter is used to specify the file system path to the JMX password file (`jmxremote.access`). The access file manages different roles/users and their access permissions. The roles managed here must match the roles in the password file. The associated value must either be "readonly" or "readwrite". Therefore, a "monitorRole" can be defined, for instance, that permits only read access to monitoring, and a "controlRole" can be defined that permits read and write access to monitoring and management.

Note on the JMX configuration of cluster nodes: Cluster nodes do not have their own configuration file, which means that all properties are loaded from the master. The JMX access to the cluster node(s) must be configured in the `fs-server.conf` of the master server. In addition, all JMX parameters for these nodes must have the prefix `cluster.<Slave-Server-Name>`. The slave server name is the name of the slave server that has been stored in the server properties in the "Clustering" area for the nodes (see Chapter 7.3.14, page 284). Example:



```
cluster.<Slave-Server-Name>.jmx.host=castor.e-spirit.com  
cluster.<Slave-Server-Name>.jmx.port=1088
```

For the parameter `cluster.<Slave-Server-Name>.jmx.host`, the fully qualified host name of the slave server or its IP address must be entered in this case.



To operate the JMX console in a production environment, user authentication and, if applicable, encrypted SSL access should always be enabled. If these parameters are disabled (default setting), access to the JMX port is not protected and unauthorized users could shut down the server via the JMX port.

For more information on configuring JMX, see:

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

4.3.1.19 Passing authentication cookies

Some HTTP servers can require user authentication to establish the connection or for communication. In this case, a cookie is used to store information (e.g. a ticket number) after successful authentication on the HTTP client (e.g. a web browser) to prevent having to re-authenticate when called repeatedly.

These authentication cookies can be used for HTTP communication between the FirstSpirit SiteArchitect and the HTTP server, i.e. the cookie is passed from the web browser via Java Web Start to a called FirstSpirit SiteArchitect.

The type of cookies to be used for passing can be defined using the parameter `clientCookieNames`.

The parameter expects as a value a comma-separated list of cookie names that are to be used for passing:

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

Updates of the cookie values by the HTTP server are accordingly taken into account for the HTTP connections.

These cookies are also passed to the browser engine used for the project and are thus available in the SiteArchitect integrated preview. The same cookie is then used in three different session contexts (web browser: HTTP client; SiteArchitect: client/server communication; SiteArchitect: integrated browser engine).





Re-injecting the cookie(s), for instance, is NOT possible when switching browsers.



The client/server communication often takes place concurrently in multiple threads, and therefore the order of cookie updating cannot be predefined. This means, for instance, that when one-time cookie management is implemented at the security infrastructure level, and it expects that new requests will always send the cookie of the previous request as well, the procedure will fail in some cases due to the concurrent processing of the client HTTP requests.

4.3.1.20 Area: SSL Parameters (FirstSpirit Version 5.1R4 and higher)

```
#####  
# SSL parameters  
#####  
# List of enabled protocols (comma separated). "DEFAULT" means use java defaults.  
# Recommended value for Java 7 or greater: TLSv1.2  
# For Java 6 only TLSv1 is supported.  
# http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#jssenames  
#  
# fs.ssl.protocols=DEFAULT  
# fs.ssl.protocols=TLSv1.2  
fs.ssl.protocols=TLSv1  
  
# List of enabled cipher suites (comma separated). "DEFAULT" means use java defaults  
(recommended).  
#  
http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#cipher suites  
#  
# fs.ssl.cipherSuites=DEFAULT
```



```
fs.ssl.cipherSuites=TLS_DH_anon_WITH_AES_128_CBC_SHA

# Client authentication parameters

# If client authentication is needed, a keystore on client side must be configured:
# -Dfs.ssl.keyStore=pathToKeystore -Dfs.ssl.keyStorePassword=123456 or
# -Djavax.net.ssl.keyStore=pathToKeystore -Djavax.net.ssl.keyStorePassword=123456

fs.ssl.wantClientAuth=false

fs.ssl.needClientAuth=false

# Jetty style OBF password obfuscation is supported for key and truststore passwords, if the
FirstSpirit specific
# parameters are used.
# http://www.eclipse.org/jetty/documentation/current/configuring-security-secure-
passwords.html

# Keystore

fs.ssl.keyStore=

fs.ssl.keyStorePassword=

fs.ssl.keyManagerPassword=

# Truststore

fs.ssl.trustStore=

fs.ssl.trustStorePassword=
```

This area provides parameters for using the SSL/TLS protocols to encrypt internal communication between the FirstSpirit server, FirstSpirit cluster nodes, FirstSpirit SiteArchitect, and the FirstSpirit web applications. By default, internal communication is encrypted without using client authentication and certificates. However, these aspects can be individually configured using the encryption parameters explained below with a view to ensuring maximum security

For more information on the "encryption of internal communication" security concept, see Chapter 4.8.1, page 148.



The following parameters can be configured in the `fs-server.conf` file:

`fs.ssl.protocols`: This parameter can be used to configure a list of valid protocol versions for internal communication. TLSv1 is selected by default. The values that can be configured here depend on which JDK is being used. If, for example, JRE 1.7 is used, TLSv1.2 can be configured here as well. If `DEFAULT` is passed, the default protocol version of the respective JRE is automatically used. Although it is also possible to use SSL, this protocol is known to contain security vulnerabilities and so is not recommended.

`fs.ssl.cipherSuites`: This parameter can be used to configure a list of valid cipher suites (standardized collections of cryptographic algorithms). Multiple entries are separated using commas. An anonymous TLS cypher suite (without a certificate) is passed by default, as there can be no assurance that all FirstSpirit installations will feature a certificate store.

Recommended configuration: To ensure secure encryption, the first step is to install a trustworthy certificate (via a certificate store) and then configure the `DEFAULT` value at this point. (Similar procedure to that described in Chapter 4.7, page 143)

`fs.ssl.keyStore`: This parameter is used to specify the path to the Java Key Store (JKS).

`fs.ssl.keyStorePassword`: This parameter is used to specify the password for the Java Key Store (JKS).

`fs.ssl.keyManagerPassword`: The JKS contains multiple private keys, each of which can have a password just like the key store itself. This parameter can be used to configure the password for the private key. In most cases, the key does not have a separate password and so it is sufficient to configure the `fs.ssl.keyStorePassword` parameter.

`fs.ssl.trustStore`: This parameter is used to specify the path to the Java Trust Store.

`fs.ssl.trustStorePassword`: This parameter is used to specify the password for the Java Trust Store.

`fs.ssl.wantClientAuth`: Client SSL authentication is disabled by default (default value: `false`). To enable client SSL authentication for internal communication over TLS, this parameter must be set to `true`. If `true` is passed here, a valid certificate for client SSL authentication is requested, but is not absolutely necessary (in contrast to the `fs.ssl.needClientAuth` parameter).

`fs.ssl.needClientAuth`: Client SSL authentication is disabled by default (default value: `false`). To enable client SSL authentication for internal communication over TLS, this parameter must be set to `true`. In this case, a valid certificate is required for successful client SSL



authentication.

Example of how to configure an encrypted connection:

```
fs.ssl.needClientAuth=true
fs.ssl.cipherSuites=DEFAULT
fs.ssl.keyStore=/home/server_cert.jks
fs.ssl.keyStorePassword=q1w2e3r4t
```

For the FirstSpirit web applications, the encryption parameters can be configured centrally on the application server (as an environment variable or a -D property, see Chapter 4.8.1.1, page 149). For cluster nodes, configuration can – once again – be performed using the `fs-server.conf` file (see Chapter 4.8.1.2, page 151). And in the case of FirstSpirit SiteArchitect, the connection settings can be used (see Chapter 4.8.1.3, page 152).

Global configuration: As an alternative to the FirstSpirit keystore parameters, the Java keystore parameters can be used instead. In this case, use of the FirstSpirit parameters is no longer permitted. In addition, the Java parameters affect all SSL instances in the Java VM (that is, they are not restricted to FirstSpirit):

```
fs.ssl.needClientAuth=true
fs.ssl.cipherSuites=DEFAULT
javax.net.ssl.keyStore=/home/server_cert.jks
javax.net.ssl.keyStorePassword=q1w2e3r4t
```

See also:

<http://docs.oracle.com/javase/7/docs/technotes/guides/security/jsse/JSSERefGuide.html#Customization>



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 ServerMonitoring (see Chapter 8.6.1.5 page 473). 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 ServerMonitoring 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 (Chapter 4.3.2.1)
- General parameters (Chapter 4.3.2.3)
- Logging (log files) (Chapter 4.3.2.4)
- System service under Windows (Chapter 4.3.2.5)

The file created during installation with default values can be found in Chapter 12.1.



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 Chapter.

Large quantities of data in FirstSpirit projects or a large number of simultaneously active FirstSpirit users 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 Clients. 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/fs-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 Oracle 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 to prevent lengthy GC pauses (-Xgcpolicy:optavgpause).

The parameters of the following Chapter 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.

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 Oracle Java VM

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

On 64-bit systems, the first step is to enable the 64-bit Java VM:

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

If only the 32bit Java VM is to be used the "-d64" parameter must be removed.

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

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



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. The utilization of this heap space over the last hour can be monitored by selecting "Monitoring -> VM memory" in the FirstSpirit Web Monitor. Normally, the amount of space occupied by FirstSpirit remains relatively constant at around 100 MB.

Java VM Garbage Collector logging on the side of the FirstSpirit Server is handled in its own file by default (`log/fs-gc.log`). The following parameters are responsible for this:

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

The `x` in `wrapper.java.additional.X=` is always a placeholder for a unique digit.

The `-Xloggc` parameter is used to change the name and storage location of the log file, if necessary. The file is automatically renamed and archived once the size defined in `conf/fs-wrapper.conf` is reached and is deleted after multiple archival steps, the number of which is also set in `conf/fs-wrapper.conf`.

Messages are written to the log file `fs-gc.log` and to a `fs-gc.*.log` file, where the name `*` includes the date and time of the first entry. If a fixed file size of 5 MB is reached, the current log file is compressed and archived to a file called `fs-gc.*.log.gz`, where again the file name includes `*` the date and time of the first entry. Archived files can be deleted or moved to `firstspirit5/backup` using the "Clean up logs" server schedule entry.

In FirstSpirit Version 5.1R4 and higher, the `-XX:+UseGCLogFileRotation` parameter can be set so that the log files rotate. In this case, the file names are based on consecutive numbering instead of being created from the date and time. This numbering starts at 0 (`fs-gc.log.0`, `fs-gc.log.1`, and so on.). With this configuration, older files are overwritten. This prevents the number of log files from constantly increasing and taking up more and more of the disk space.

The `-XX:GCLogFileSize` parameter can be used to set the maximum file size that can be reached before a new file is created. The minimum value is 8 KB.



The `-XX:NumberOfGCLogFiles` parameter can be used to set the maximum number of log files that should be retained.

Example:

```
wrapper.java.additional.X=-XX:+UseGCLogFileRotation
wrapper.java.additional.X=-XX:GCLogFileSize=10M
wrapper.java.additional.X=-XX:NumberOfGCLogFiles=9
```

If `-XX:+UseGCLogFileRotation` is set, the following message is recorded in the `fs-server.log` log file (inside the `~/log/` directory) when the FirstSpirit Server is started:

```
INFO 17.12.2014 13:26:35.694
(de.espirit.firstspirit.server.logging.GcLogTailer): Fs-GcLogTailer not
started. Either no gc log has been configured (param -Xloggc:) or vm
internal log rotation is being used (param -XX:+UseGCLogFileRotation)
```

For more information on the individual parameters, see <http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html>.

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.

The following parameters also activate logging of the Garbage Collector invocations in the `log/fs-gc.log` file (see also Chapter 4.3.2.2.1 page 76):

```
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/jdk1.7.0/bin/java` or `c:\JDK1.7.0\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. The value which is defined here can be recognized also in the graphic about the memory consumption in the FirstSpirit ServerMonitoring (see Figure 8-1 and description in Chapter 8.1.1).



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). As long as the parameter `wrapper.ignore_sequence_gaps` is set to `true` the numbering need not to be consecutive.

`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`.

The default value für this parameter is `0027`. This means that the user class "group" has read permissions and the user class "others" has no permissions at all. Cf. also

http://en.wikipedia.org/wiki/Filesystem_permissions#Traditional_Unix_permissions.

The mask that controls which file permissions are set for files and directories for the Unix system is overwritten by the mask that controls the permissions which are set for FirstSpirit by using the parameter `wrapper.umask`. This means, that permissions which are defined for the operating system do potentially not apply to files and directories which are created by FirstSpirit. For this reason, the value of the parameter `wrapper.umask` should be checked and adapted to the permission mask of the system, if necessary.

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 "fs5 console" under Unix or via `Start menu → FirstSpirit 5.1 → 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 Chapter 4.3.6), this log is also sent to syslogd instead of `fs-server.log`.

The FirstSpirit Server logs one time when starting and cyclically each hour the VM `StartTime` and the VM `Uptime`:



Suchergebnis (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 ServerMonitoring: Log VM StartTime and VM Uptime



Log messages which are amongst others relevant for debugging can be found not only in the file `fs-wrapper.log` but also in the file `fs-server.log` (see Chapter 4.3.6 Seite 103).

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 `Start menu → FirstSpirit 5.1 → Installation → Deregister as service / Register as service`. The FirstSpirit Server is stopped and started for this task.

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

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

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

`wrapper.nts.service.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 unique numbering (X) for each new service.

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

`wrapper.nts.service.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  
wrapper.ignore_sequence_gaps
```



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 ServerManager (see Chapter 7.3.5 page 252). 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 ServerManager.

```
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 Chapter 4.9 page 155 for further information on configuring the database connection. The parameters are described in the Chapters

- 4.9.4.1: obligatory parameters
- 4.9.4.2: optional parameters
- 4.9.4.3: oracle-specific parameters and
- 4.9.4.4: MS-SQL specific parameters

from page 169 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 ServerManager (see Chapter 7.3.10 page 264) or via ServerMonitoring (see Chapter 8.6.1.8 page 476). 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 Chapter 12.2.



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.

FirstSpirit uses the Java standard JAAS⁷ 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.

⁷ Java Authentication and Authorization Service:

<http://docs.oracle.com/javase/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 Chapter 4.3.1.10. 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



mentioned 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 Microsoft Internet Explorer is 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:
`":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".

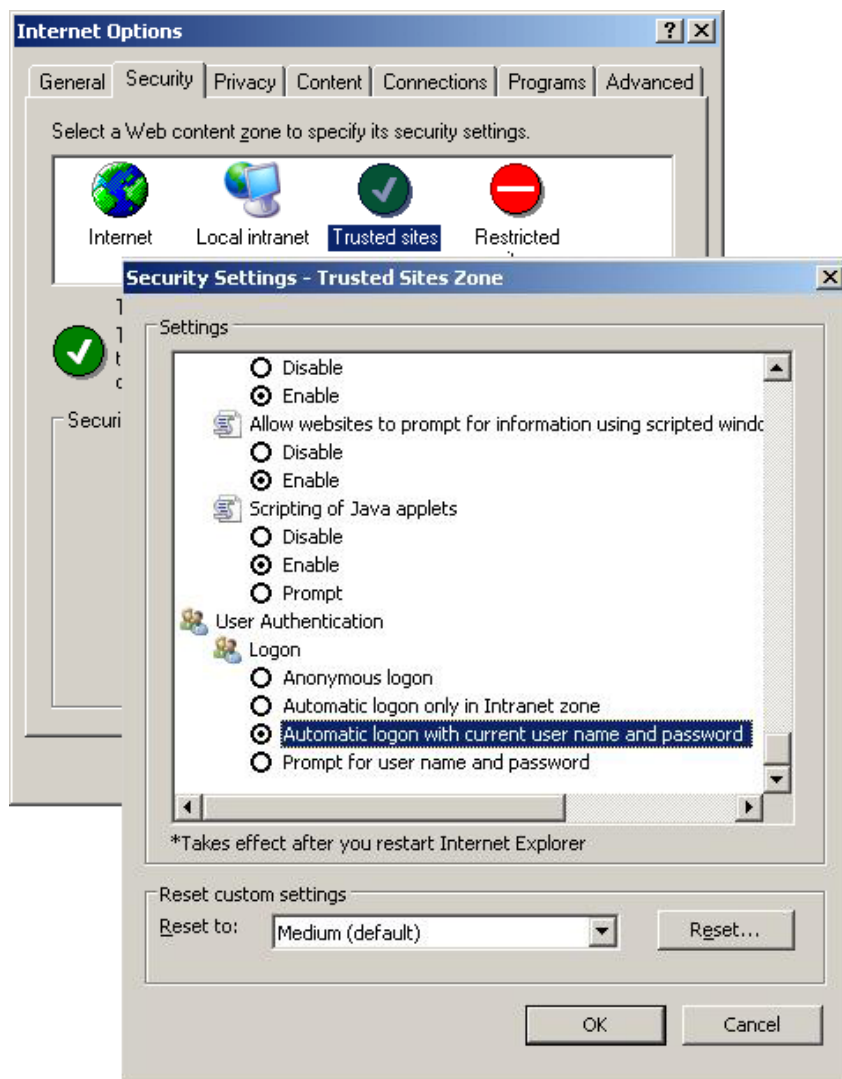


Figure 4-2: Adapt security settings



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 ":

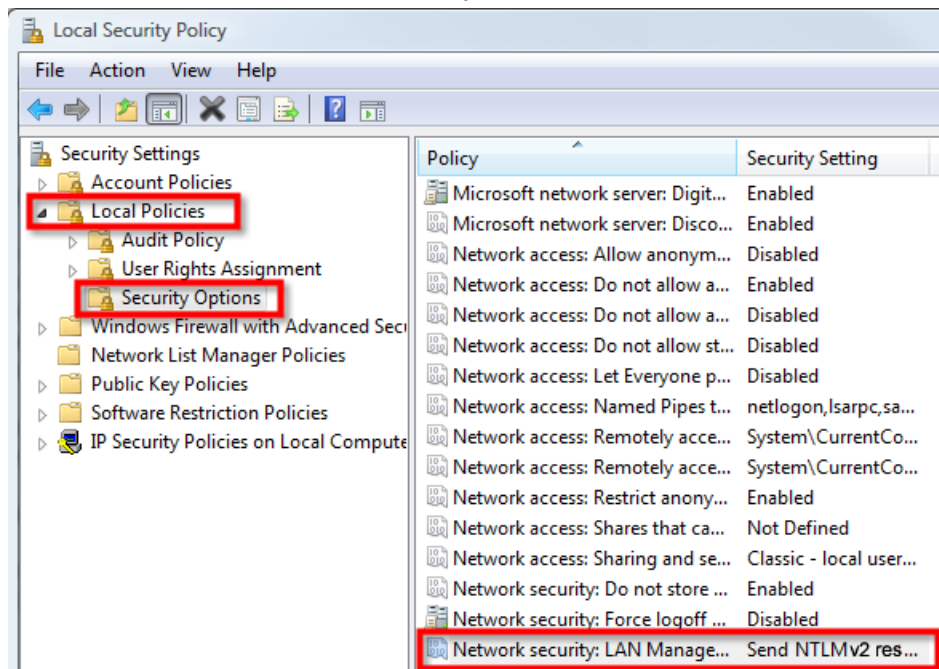


Figure 4-3: Network security: LAN manager authentication level

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):



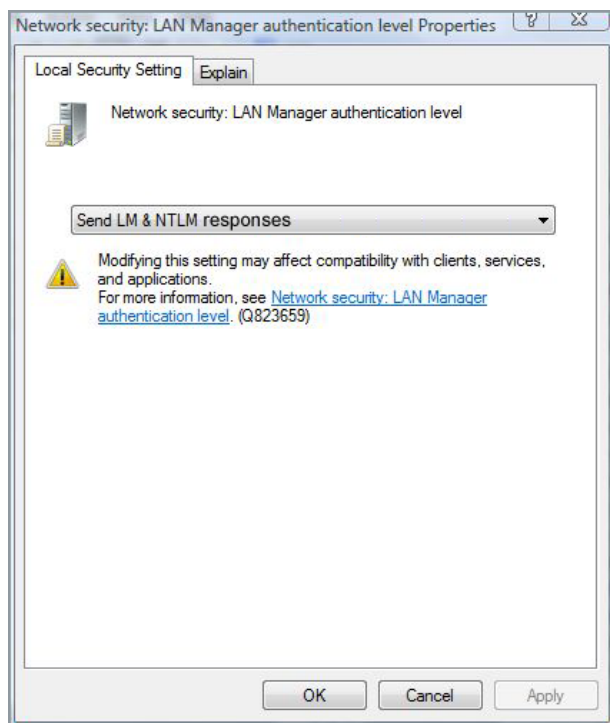


Figure 4-4: Default setting in Windows XP

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



In addition to the NTLM login module, the Kerberos login module is also available (see Chapter 4.3.4.5 page 90). 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)

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/fs5host.mydomain.net@MYDOMAIN.NET"
  keyTab="/opt/firstspirit5/conf/fs5host-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 Chapter.



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 Tools⁸ 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, fs5host-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/fs5host.mydomain.net@MYDOMAIN.NET \  
+rndpass -mapuser fs5host-HTTP \  
-crypto RC4-HMAC-NT -ptype KRB5_NT_PRINCIPAL \  
-out fs5host-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 PASSWORD.

⁸ 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/fs5host.mydomain.net@MYDOMAIN.NET \  
+rndpass -mapuser fs5host-HTTP \  
-crypto AES256-SHA1 -ptype KRB5_NT_PRINCIPAL \  
-out fs5host-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 fs5host-HTTP
```

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

```
setspn -d HTTP/fs5host.mydomain fs5host-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/firstspirit5/conf/fs5host-HTTP.keytab`.

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

```
kinit -V -k -t fs5host-HTTP.keytab HTTP/fs5host.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-fs5host-HTTP-rc4.keytab  
rkt krb5-fs5host-HTTP-aes256.keytab  
wkt /opt/firstspirit5/conf/fs5host-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/firstspirit5/conf/fs-jaas.conf
```

The use of Kerberos tickets increases the size of the HTTP request header by a few kilobytes, exceeding the default configuration with regard to the usual maximum length of only 4 kilobytes for some web servers. Here is how to configure the different web servers in order to increase the maximum length of the HTTP request:

Jetty (integrated in FirstSpirit):

In the `/opt/firstspirit5/conf/fs-webapp.xml` file of the `org.mortbay.jetty.nio.SelectChannelConnector` section, add the following parameter:

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

Tomcat:

In the `tomcat/conf/server.xml` file of the `"<Connector protocol="HTTP/1.1""` section, add the following parameter:

```
maxHttpHeaderSize="65536"
```

and in the `"<Connector protocol="AJP/1.3""` section, add the following parameter:

```
packetSize="65536"
```

Apache httpd:

In the `httpd.conf` file, or in the equivalent file on virtual hosts, add the following parameter:

```
LimitRequestLine 65536
```

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/firstspirit5/log/fs-server.log` and

`/opt/firstspirit5/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/firstspirit5/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 its 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.

If problems occur when setting up the Kerberos-based login, you can start Firefox with the debugging log to find out which ticket the browser is sending.

Windows:

Launch the command prompt (`cmd.exe`) and enter:

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

Unix:

Open the terminal window (`konsole`, `xterm`, or similar) and enter:

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



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.

For more information about the configuration of the SAPLoginModule please see also module documentation "SAP Business Package for FirstSpirit".

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 SiteArchitect unless it is started via Java Web Start. The NTLMLoginModule (see Chapter 4.3.4.4 page 86) is sufficient for the SiteArchitect 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.



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 SiteArchitect, ContentCreator, 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 installation 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:

- **SiteArchitect:** `JAAS.client`
- **ServerManager:** `JAAS.admin`
- all FirstSpirit web applications (ContentCreator, start page, ServerMonitoring) with SSO authentication: `JAAS.websso`
- all FirstSpirit web applications (ContentCreator, start page, ServerMonitoring) without SSO 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 Chapter 12.2 the following login method results for the SiteArchitect:

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 ServerManager. After successful authentication the ticket is generated and transferred by the web browser at a later date.
2. When starting the SiteArchitect via Web Start, the web browser transfers the previously created ticket via the SiteArchitect 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 SiteArchitect 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.NTLMLLoginModule optional
    domains="E-SPIRIT:dc1.e-spirit.de,dc2.e-spirit.de;dc1.e-spirit.de,dc2.e-spirit.de";
  de.espirit.firstspirit.server.authentication.LdapLoginModule optional section="LDAP";
  de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

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

```

Extract from the file `/opt/firstspirit5/conf/fs-server.conf`:

```

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

```



4.3.5 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 ServerMonitoring (see Chapter 8.6.1.2 page 471). Furthermore, it is also possible to insert a new licence file via ServerMonitoring. 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
#Mon Jan 02 10:34:22 CET 2012
license.MAXPROJECTS=5
license.EXPDATE=15.12.2012
license.MODULES=personalisation,search,integration,newsletter,portal,form_ed
it,enterprise_search
license.ARCHIVE=1
license.VERSION=5
license.USER=e-spirit
license.WORKFLOW=1
license.DOCUMENTGROUP=1
license.MAXSESSIONS=20
license.WEBEDIT=1
license.MAXUSER=20
license.OFFICE_INTEGRATION=1
license.APPTAB_SLOTS=5
-----begin FirstSpirit license key-----
...
-----end FirstSpirit license key-----
```

`license.MAXPROJECTS`: Maximum number of projects which can be created with this licence on the server. Deactivated projects are not 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 Chapter 4.3.1.9 page 51), 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 licensee.

`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 Chapter 7.9 page 443) 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`: This value is set to “1” by default. Thus, the “Document group” function is activated, 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 Online Documentation / Advanced topics / Document Groups* 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 opened) to terminate current sessions, if necessary.

`license.WEBEDIT`: Value “1” activates the “ContentCreator” function, thus enabling the direct editing of editorial contents within the preview page in the browser window.

`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 SiteArchitect by using a correspondent logo. In the case of a “Productive” licence the project logo which can be selected in the ServerManager for the respective project (see Chapter 7.4.2 page 294) 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 (SiteArchitect and ContentCreator) 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 documentation *FirstSpirit AppCenter*.)

For further licence keys please see respective module documentation.

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 ServerMonitoring (see Chapter 8.6.1.3 page 472). 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”⁹. 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.

⁹ Further information <http://logging.apache.org/log4j/docs/documentation.html>



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 ServerMonitoring during runtime (see Chapter 8.6.1.3 page 472).

Configuration files should have the following syntax:

`fs-logging.myLogging.conf` to ensure they are recognised by FirstSpirit and activated via ServerMonitoring.

```
log4j.rootCategory=INFO, fs

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

# fs
log4j.appender.fs=de.espirit.firstspirit.server.logging.FSAppender
log4j.appender.fs.consoleLogging=false
log4j.appender.fs.plainLogging=false
log4j.appender.fs.datedLogging=true
log4j.appender.fs.maxFileSize=5MB
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:/FirstSpirit5/log/err.log
```

Example:

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

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 ServerMonitoring 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.



Log messages which are amongst others relevant for debugging can be found not only in the file `fs-server.log` but also in the file `fs-wrapper.log` (see Chapter 4.3.2.4 page 81).

4.3.7 Web server configuration (fs-webapp.xml)

FirstSpirit provides an integrated web server including servlet engine for preview creation, ServerManager and working with ContentCreator; this web server is automatically configured and activated during installation. Jetty¹⁰ 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 Chapter 4.5 page 113.

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 ServerMonitoring (see Chapter 8.6.1.6 page 474). The changes are then written into 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 Chapter 12.3.

¹⁰ Further information <http://www.mortbay.org/>





Changes in the configuration file will only become effective after restarting the web server via FirstSpirit ServerMonitoring (see Chapter 8.6.2.2 page 479).

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 Chapter 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 Chapter 4.3.1.14 for configuration of the



FirstSpirit session timeout.

Further possible connectors are AJP (Chapter 4.5) and HTTPS (Chapter 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_WEBEDIT5_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBEDIT5_URL" /></Arg>
</New>
<New class="org.mortbay.jetty.webapp.WebAppContext">
    <Arg><Ref id="Server" /></Arg>
    <Arg><SystemProperty name="WEBAPP_STAGING_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_STAGING_URL" /></Arg>
</New>
```



4.3.7.3 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 (Chapter 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`¹¹ for redirecting.

¹¹ rinetd: <http://www.boutell.com/rinetd/>



4.4 Connection to an LDAP server

4.4.1 Authentication via LDAP

In FirstSpirit you have the option of using an LDAP¹² server to handle user authentication. A user is assigned a flag, which shows if he is or is not an external user. Different LDAP configurations (called sections) can be created (see 4.3.4 page 85) and configured (see 4.3.1.10 page 52) in FirstSpirit Server. An LDAP user is associated with one section only (see 7.2.4.2 page 235). Authentication can potentially take place three different ways:

1. **LDAP Bind:** the name and password are sent to the LDAP server. The "Distinguished Name" (DN), i.e. the unique user identification key, must be known within the LDAP server. If the DN exists, the password passed is checked using the "Bind" operation. An example of using LDAP Bind is available in Chapter 4.3.1.10 page 52.
2. **LDAP Search & Bind:** if the "Distinguished Name" (DN) of a user is unknown, you can search for it within a subtree of the LDAP server. A search filter and start node must be defined to do 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" is the same as the login name entered. The start node is the node with the DN "dc=mycompany,dc=com". If this type of node is found, a "Bind" is executed (see LDAP Bind).

3. **LDAP Search & Compare:** the function of this option is equivalent to option 2. However, after a matching node is found, no "Bind" operation is carried out. Instead, the password entered is compared to any LDAP attribute desired.

Example:

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

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

¹² LDAP (Lightweight Directory Access Protocol)



Once LDAP authentication is successful, the user is added to the FirstSpirit server as an external user if the user was previously unknown to the FirstSpirit system (see Chapter 7.4.7.2 page 314). To do this, the configuration parameter `JAAS.autoCreateUser` must be configured to the value `true` (default setting) (see Chapter 4.3.4.8 page 97). The external LDAP user login is automatically copied to FirstSpirit. The **external LDAP user** password, however, is cleared after it is successfully authenticated for the first time in FirstSpirit. Logins with an empty password are rejected by the FirstSpirit server. An external LDAP user can therefore only log in to the FirstSpirit server when the LDAP server is available during the login procedure. In the case of **internal LDAP users**, the password is retained in FirstSpirit. The user in this case can log in using the FirstSpirit password and the LDAP password.

The **server administrator** (login: Admin) has a special role for LDAP login. The user is created automatically when the FirstSpirit server is installed. The server administrator password is never cleared, regardless of whether the user has been configured as "external" or "internal". It is therefore recommended that the server administrator password (initially "Admin") be changed immediately after FirstSpirit Server has been installed.

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 Chapter 4.3.1.10 page 52).

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 Chapter 4.3.1.10 page 52 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`¹³. 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` (Chapter 4.3.1.1):

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

¹³ <http://www.openssl.org/>





The following applies to all `wrapper.java.additional.` parameters: only one Java parameter per line. All specified Java parameters have to contain consecutive, unique numbering (X). As long as the parameter `wrapper.ignore_sequence_gaps` is set to true the numbering need not to be consecutive.*

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 in the FirstSpirit server by default as the HTTP server and servlet engine. If FirstSpirit projects are to use special, server-side implementations (e.g. PHP or ASP) that cannot be evaluated by Jetty, an additional external web server must be integrated. The FirstSpirit web applications in the internal Jetty web server then forward the HTTP queries for specific file types via HTTP to the external web server.

This Chapter describes the use of the Apache HTTP server in combination with PHP. Other web servers can be integrated using the same principle, as long as they allow forwarding to the servlet engine via HTTP or AJP.

The Jetty 8 web server integrated in FirstSpirit 5 features the Servlet 3.0 and JSP 2.1 standards and is not fully compatible with AJP. If web applications in FirstSpirit projects require the complete J2EE standard or AJP to connect an HTTP server or load balancer, Apache Tomcat is therefore available for this purpose. See Chapter 4.5.2 for more information.

Another configuration option is to balance the load across multiple servlet engines (see Chapter 4.5.5 page 137).



4.5.1 Apache HTTP Server with the Jetty servlet engine

Apache HTTP Server version 2.2 or 2.4 is used in this configuration in conjunction with the Jetty web server servlet/JSP engine integrated in FirstSpirit. Since the Jetty web server is not fully compatible with AJP 1.3, HTTP must be used in this case as the protocol between Apache and Jetty.

To establish the HTTP connection to the servlet engine, `mod_proxy_http`¹⁴ is implemented.

The Apache configuration environment depends on the operating system and is usually distributed across multiple configuration files. A standard convention is to use the file `/etc/apache2/httpd.conf` for general parameters, to use the directory `/etc/apache2/mods-available` for the module configuration, and to use a file for each virtual web server under `/etc/apache2/sites-available`.

FirstSpirit should have its own virtual web server that is configured using the following entries so that HTTP queries for the FirstSpirit web applications are forwarded to Jetty via `mod_proxy_http`. From the FirstSpirit web applications installed under Jetty, an internal HTTP connection is then also established automatically, if necessary, for delivery of specific file types such as PHP or ASP.

Check the FirstSpirit Jetty configuration before configuring Apache. The `firstspirit5/conf/fs-server.conf` file should not contain the parameters `WEBAPP_ROOT_PATH` and `WEBAPP_ROOT_URL`; the following parameter must be defined:

```
INTERNAL_SERVLET_ENGINE=1
```

Apache can now be configured:

In the configuration example, the lines labeled "EXAMPLE" must be adapted to point to the paths and addresses for the local configuration. The entry "# EXAMPLE" must then be deleted; otherwise a syntax error will be displayed when the web server is started.

After the Apache server is configured and restarted, the FirstSpirit start page (example: `http://fs5.yourdomain.net`) can be accessed via Apache.

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



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 module configuration:

```
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 fs5.yourdomain.net          # EXAMPLE
  ServerAlias fs5                        # EXAMPLE

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

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

  DocumentRoot /opt/firstspirit5/web          # EXAMPLE

  DirectoryIndex index.html index.jsp index.php

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

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

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

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

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

</VirtualHost>

```



If the start page is to be displayed directly via Jetty under the Apache httpd environment for testing or for administration purposes, <http://fs5server:8000> needs to be entered as the start page in the browser. Port 8000 in this case corresponds to the Jetty server port for `HTTP_PORT` entered in `fs-server.conf`.

To use HTTPS in the Apache httpd and Jetty combination, the following modifications are necessary:

In the Apache configuration, additional information must be added to the following parameters:

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

The file path specified 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 `firstspirit5/conf/fs-webapp.xml` file, enable the "HTTPS Connector" area and enter 8443 for the `port`. The supplied self-signed certificate already registered in the configuration in the `conf/fs-keystore.jks` keystore can easily be used for this configuration because Jetty HTTPS Connector is used only internally between Apache httpd and Jetty.



4.5.2 Apache HTTP Server with the Tomcat servlet engine

Apache HTTP Server version 2.2.22¹⁵ or later is used in this configuration in conjunction with the Tomcat 6 or 7 servlet engine. AJP is used as the protocol between Apache and Tomcat.

The Jetty 8 web server integrated in FirstSpirit 5 offers the Servlet 3.0 and JSP 2.1 standards and is not fully compatible with AJP. If web applications in FirstSpirit projects require the complete J2EE standard or AJP to connect an HTTP server or load balancer, Apache Tomcat is therefore available for this purpose.

Currently, `mod_jk` and `mod_proxy_ajp`¹⁶ are available as AJP connectors for Apache. This Chapter describes the use of `mod_proxy_ajp`, since this module has been included with the software since version 2.2 and thus simplifies installation compared to `mod_jk`. Load balancing across multiple servlet engines is possible with both modules (see Chapter 4.5.5 page 137).

The Apache configuration environment depends on the operating system and is usually distributed across multiple configuration files. A standard convention is to use the file `/etc/apache2/httpd.conf` for general parameters, to use the directory `/etc/apache2/mods-available` for the module configuration, and to use a file under `/etc/apache2/sites-available` for each virtual web server.

FirstSpirit should have its own virtual web server that is configured using the following entries so that HTTP queries are forwarded to Tomcat for FirstSpirit web applications via `mod_proxy_ajp`. From the FirstSpirit web applications installed under Tomcat, an internal HTTP connection is then also made automatically, if necessary, for delivery of specific file types such as PHP or ASP.

In the configuration example, the lines labeled "EXAMPLE" must be adapted to point to the paths and addresses for the local configuration. The entry "# EXAMPLE" must then be deleted; otherwise a syntax error will be displayed when the web server is started.

¹⁵ Due to a bug in `mod_proxy_ajp`, which was fixed in version 2.2.22, it is recommended that Apache `httpd` version 2.2.22, 2.4 or later be used with FirstSpirit. If a faulty version of `mod_proxy_ajp` is used, a single HTTP request whose processing time exceeds the timeout value specified in `httpd.conf` will cause disconnection of the entire Tomcat worker. A workaround for most of the problems in older versions of `mod_proxy_ajp` is to use a value of sufficient size for the "timeout" parameter, e.g. 1200s, as in the following configuration text.

¹⁶ http://httpd.apache.org/docs/2.2/mod/mod_proxy.html



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_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 fs5.yourdomain.net          # EXAMPLE
ServerAlias fs5                        # EXAMPLE

LogLevel warn

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

CustomLog /var/log/apache2/fs5.access.log route_and_requesttime # EXAMPLE
ErrorLog /var/log/apache2/fs5.error.log          # EXAMPLE

ServerSignature off
UseCanonicalName off
AddDefaultCharset off
ProxyRequests off
RewriteEngine on

DocumentRoot /opt/firstspirit5/web          # EXAMPLE

DirectoryIndex index.html index.jsp index.php

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

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

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

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

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

</VirtualHost>

```



Configuration for a Tomcat servlet engine that will run on the same host as FirstSpirit Server is described in the next Chapter (4.5.3), and configuration for a Tomcat servlet engine that will run on its own host is described under 4.5.4.

4.5.3 Tomcat servlet engine

The Jetty 8 web server integrated in FirstSpirit 5 features the Servlet 3.0 and JSP 2.1 standards and is not fully compatible with AJP. If web applications in FirstSpirit projects require the complete J2EE standard or AJP to connect an HTTP server or load balancer, Apache Tomcat 6¹⁷ or 7 should be used. The following Chapter describes how to configure it for FirstSpirit.



For supplementary information about Tomcat version 7.0.42 and higher please refer to:

<https://community.e-spirit.com/docs/DOC-1758>

Tomcat can run on the same host as FirstSpirit Server (covered in this Chapter) or on its own host (see 4.5.4) or on multiple hosts for load balancing (Chapter 4.5.5).

Tomcat runs in this configuration on the same host as FirstSpirit Server and requires read and write access to the directory `firstspirit5/web`. Since read and write access to the same folder is required by FirstSpirit Server as well, Tomcat needs to run under the same user account as FirstSpirit Server, i.e. in the "fs5" user account installed by default.

If for security reasons you need to run Tomcat under a different user account, use the option described in Chapter 4.5.4, which can also be used on the same host.

Tomcat can either be run on its own with the HTTP server integrated in Tomcat, or it can be enhanced to include the Apache HTTP Server, as described in Chapter 4.5.2.

The FirstSpirit configuration must be modified before configuring Tomcat. The following parameters must be defined in the `firstspirit5/conf/fs-server.conf` file in order to

¹⁷ <http://tomcat.apache.org>. A bug (https://issues.apache.org/bugzilla/show_bug.cgi?id=50700) in Tomcat versions 6.0.30 through 6.0.32 prevents reading out the context parameters required for FirstSpirit which are located in the `context.xml` file. Either Tomcat version 6.0.29 or at a minimum version 6.0.33 or 7.0 should therefore be used.



disable Jetty and to define the default web application to be used when the FirstSpirit start page is called via Tomcat (e.g. `http://fs5.yourdomain.net:8080`):

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

Then shut down FirstSpirit Server, delete the directory `firstspirit5/web/fs5root` and restart FirstSpirit Server.

In this example, Tomcat version 6 or 7, using the "Binary Distribution Core" from `http://tomcat.apache.org`, is installed to `/opt/firstspirit5/tomcat`. If it is installed to a different directory, the relative path `"${catalina.home}/../"` at all locations in the following example configuration must be replaced with the absolute path `"/opt/firstspirit5/"`.

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

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

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

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

In the `tomcat/conf/web.xml` file, enable checking and compiling of JSP files immediately after each change. To do this, replace the 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>
```



with the following:

```
<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>
```

If the option of listing the directory contents for the development of web applications in the FirstSpirit server staging area is required, replace lines 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>>false</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

with the following:

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

Tomcat 6:

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

<tomcat-users>

  <role rolename="manager"/>

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

</tomcat-users>
```

Tomcat 7:

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

<tomcat-users>

  <role rolename="manager-gui"/>

  <role rolename="manager-script"/>

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

</tomcat-users>
```

Create the file `tomcat/conf/Catalina/localhost/manager.xml` with the following content to activate 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`¹⁸ with the following content to enable careful checking for file changes (HTML). FirstSpirit Server must be accessible from Tomcat within the local network via the host name specified for "firstspirit.host" or the IP address. The "firstspirit.port" entry must match the FirstSpirit Server port, as defined in `firstspirit5/conf/fs-server.conf` by the `SOCKET_PORT` parameter. In addition, saving persistent session information is disabled because the FirstSpirit web applications do not support this:

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

<Context allowLinking="true" cachingAllowed="false" useHttpOnly="true">

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

  <!-- disable session persistence across Tomcat restarts -->

  <Manager pathname="" />

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

</Context>
```

In `tomcat/conf/catalina.properties`, expand the FirstSpirit installation entry "common.loader" to include `fs-webrt.jar` and `shared/lib` (**type everything on one line without spaces**):

```
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}` a different path will be used under Windows, the following lower-case drive letters should be used (**type everything on one line without spaces**):

```
common.loader=${catalina.home}/lib,${catalina.home}/lib/*.jar,

d:/Programme/FirstSpirit5/data/fslib/fs-webrt.jar,

d:/Programme/FirstSpirit5/shared/lib/*.jar
```

¹⁸ A bug (https://issues.apache.org/bugzilla/show_bug.cgi?id=50700) in Tomcat versions 6.0.30 through 6.0.32 prevents reading out the context parameters required for FirstSpirit which are located in the `context.xml` file. Either Tomcat version 6.0.29 or at a minimum version 6.0.33 or 7.0 should therefore be used.



Create the file `tomcat/lib/log4j.properties` with the following content in order to redirect logging of FirstSpirit web applications to a separate log 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.eclipse.jetty=WARN
log4j.logger.org.apache.catalina=INFO
log4j.logger.org.apache.jasper=WARN
log4j.logger.org.apache.log4j.jmx=ERROR
log4j.logger.org.apache.commons.httpclient=INFO

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

When using Tomcat 6, download the `log4j-1.2.*.jar` file from <http://logging.apache.org/log4j/1.2/download.html> and copy it to `tomcat/lib/`. This step is not necessary when using Tomcat 7.



When using Tomcat instead of Jetty, at all points in the `ServerManager` (in the server properties, see Chapters 7.3.12 and 7.3.13, starting on page 271, and Chapter 7.4.18 page 343 in the project properties) as well as in the `fs-server.conf` file, "InternalJetty" appears everywhere as the web server identifier, since from the perspective of FirstSpirit Server, the Tomcat server uses exactly the same files at the same locations as the Jetty web server.

The application server configuration requirements described in Chapter 4.6.3 (page 141) apply to the Java VM used by Tomcat. For Tomcat, configuration is handled in the file `tomcat/bin/setenv.sh` with the following content, which corresponds to the JVM parameters from `firstspirit5/conf/fs-wrapper.conf`. The input for the Java heap size (Xmx, Xms, Xmn) must be adjusted according to the available RAM.

```
#export JAVA_OPTS="-Xmx3072m"
#export LC_CTYPE=de_DE.UTF-8
#CATALINA_OPTS="-Djava.security.auth.login.config=/opt/firstspirit5/conf/fs-jaas.conf"

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



```

# Tomcat Heapsize settings
# set Xmx and Xms to max of 75% of available RAM, max 10000M
# set Xmn to 40% of Xmx
# Change jmxremote port to any free availble port
# and consider activiated jmx password security.
CATALINA_OPTS="\
-Xmx4096M -Xms4096m -Xmn1664m \
-XX:PermSize=500m -XX:MaxPermSize=500m \
-XX:InitialCodeCacheSize=128m \
-XX:ReservedCodeCacheSize=128m \
-XX:SurvivorRatio=1 \
-XX:SoftRefLRUPolicyMSPerMB=1 \
-XX:+NeverTenure \
-XX:-UseLargePages \
-XX:+UseParNewGC \
-XX:+UseConcMarkSweepGC \
-XX:+CMSParallelRemarkEnabled \
-XX:+CMSClassUnloadingEnabled \
\
-Djava.awt.headless=true \
-Dfile.encoding=UTF-8 \
-Djava.net.preferIPv4Stack=true \
-Djava.security.auth.login.config=/opt/firstspirit5/conf/fs-jaas.conf \
-Djava.security.policy=conf/fs-server.policy \
-Xshare:off \
-Djava.net.preferIPv4Stack=true \
-Djava.io.tmpdir=work \
\
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.ssl=false \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.port=8006 \
\
-verbose:gc \
-XX:+PrintGCTimeStamps \
-XX:+PrintGCDateStamps \
-XX:+PrintGCDetails \
-Xloggc:/opt/tomcat/logs/tomcat-gc.log \
"
CATALINA_PID=/opt/firstspirit5/tomcat/work/catalina.pid

```

For information about switching to a GarbageCollector log file rotation please see Chapter 4.3.2.2.1 page 76.

In the file firstspirit5/tomcat/bin/catalina.sh, change the line

```
FORCE=0
```

to

```
FORCE=1
```

so that you can define how to shut down Tomcat without "hanging". If you do not want catalina.sh



modified, the parameter "-force" must be added, i.e. "catalina.sh stop -force", each time Tomcat is stopped.

Tomcat is now started using the following line:

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

To stop:

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

To start Tomcat automatically at the same time as FirstSpirit at system startup, add the following line to /etc/init.d/fs5 in the "start" area

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

and the following line in the "stop" area):

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



4.5.4 Tomcat servlet engine on a dedicated host

The Jetty 8 web server integrated in FirstSpirit 5 features the Servlet 3.0 and JSP 2.1 standards and is not fully compatible with AJP. If web applications in FirstSpirit projects require the complete J2EE standard or AJP to connect an HTTP server or load balancer, Apache Tomcat 7¹⁹ should be used. The following Chapter describes how to configure it for FirstSpirit.

Tomcat can run on the same host as FirstSpirit Server (see Chapter 4.5.3) or on its own, dedicated host, (covered in this Chapter), or on multiple hosts for load balancing (see Chapter 4.5.5).



For supplementary information about Tomcat version 7.0.42 and higher please refer to:

<https://community.e-spirit.com/docs/DOC-1758>

In this configuration, Tomcat runs on a host that is different from that of the FirstSpirit server host in order to provide a better load balance with regard to CPU and RAM resources.

Tomcat can either be run on its own with the HTTP server integrated in Tomcat or it can be enhanced to include the Apache HTTP Server, as described in Chapter 4.5.2.

The FirstSpirit configuration must be modified before configuring Tomcat. The following parameters must be added to the `firstspirit5/conf/fs-server.conf` file in order to define the default web application to be used when calling the FirstSpirit start page via Tomcat (`http://fs5.yourdomain.net:8080` in this example) and to define the URL over which Tomcat is to be accessible from the perspective of the clients in order to generate a valid URL when sending messages outside of active client connections:

¹⁹ <http://tomcat.apache.org>. A bug (https://issues.apache.org/bugzilla/show_bug.cgi?id=50700) in Tomcat versions 6.0.30 through 6.0.32 prevents reading out the context parameters in the `context.xml` file that are required for FirstSpirit. Either Tomcat version 6.0.29 or at a minimum version 6.0.33 or 7.0 should therefore be used.



```
WEBAPP_ROOT_PATH=${WEB_DIR}/ROOT

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

If HTTPS is used, or if socket mode is to be used instead of the FirstSpirit client HTTP connection mode, refer to the extended parameter for "fs.url" in Chapter 4.3.1.1.

Then shut down the FirstSpirit server, delete the directory `firstspirit5/web/fs5root` and restart the FirstSpirit server.

In this example, Tomcat version 6 or 7, using the "Binary Distribution Core" from <http://tomcat.apache.org>, is installed under `/opt/tomcat`.

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

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

In the `tomcat/conf/web.xml` file, enable checking and compiling of JSP files immediately after each change. To do this, replace the 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>
```

with the following:



```
<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>
```

If the option of listing the directory contents for the development of web applications in the FirstSpirit server staging area is required, replace lines 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>>false</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

with the following:

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



Tomcat 6

To enable status monitoring in Tomcat Manager, insert a user account for the "manager" role in the Tomcat Manager `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>
```

Tomcat 7

To be able to use Tomcat Manager for status monitoring via the URL `http://fs5.yourdomain.net:8080/manager/html` as an "Admin" user and to automate deployment of web apps via FirstSpirit, add a user account for each Tomcat Manager "manager-gui" and "manager-script" role to the `tomcat/conf/tomcat-users.xml` file:

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

<tomcat-users>

  <role rolename="manager-gui"/>

  <role rolename="manager-script"/>

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

  <user username="fsdeploy" password="deploy-password" roles="manager-script"/>

</tomcat-users>
```

For supplementary information please refer to <https://community.e-spirit.com/docs/DOC-1772>.

Replace the existing file `tomcat/conf/context.xml`²⁰ with the following content to enable careful checking for file changes (HTML). FirstSpirit Server must be accessible from Tomcat within the local network via the host name specified for "firstspirit.host" or the IP address. The "firstspirit.port" entry must match the FirstSpirit server port, as defined in `firstspirit5/conf/fs-`

²⁰ A bug (https://issues.apache.org/bugzilla/show_bug.cgi?id=50700) in Tomcat versions 6.0.30 through 6.0.32 prevents reading out the context parameters in the `context.xml` file that are required for FirstSpirit. Either Tomcat version 6.0.29 or at a minimum version 6.0.33 or 7.0 should therefore be used.



server.conf by the SOCKET_PORT parameter. In addition, saving persistent session information is disabled because the FirstSpirit web applications do not support this:

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

<Context allowLinking="true" cachingAllowed="false" useHttpOnly="true">

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

    <!-- disable session persistence across Tomcat restarts →

    <Manager pathname="" />

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

</Context>
```

Create the file `tomcat/lib/log4j.properties` with the following content in order to redirect logging of FirstSpirit web applications to a separate log 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
```

Download the `log4j-1.2.*.jar` file from <http://logging.apache.org/log4j/1.2/download.html> and copy it to `tomcat/lib/`. *This step is only necessary when using Tomcat 6 and is not required when using Tomcat 7.*

The application server configuration requirements described in Chapter 4.6.3 (page 141) apply to the Java VM used by Tomcat. For Tomcat, configuration is handled in the file `tomcat/bin/setenv.sh` with the following content, which corresponds to the JVM parameters



from firstspirit5/conf/fs-wrapper.conf. The input for the Java heap size (Xmx, Xms, Xmn) must be adjusted according to the available RAM.

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

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

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

For information about switching to a GarbageCollector log file rotation please see Chapter 4.3.2.2.1 page 76.

In the file firstspirit5/tomcat/bin/catalina.sh, change the line

```
FORCE=0
```

to



```
FORCE=1
```

so that you can define how to shut down Tomcat without "hanging". If you do not want catalina.sh modified, the parameter "-force" must be added, i.e. "catalina.sh stop -force", each time Tomcat is stopped.

Tomcat is now started using the following line:

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

To stop:

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

To launch Tomcat automatically upon system startup, create the file /etc/init.d/tomcat with the following content, define file access rights using `chmod a+rx /etc/init.d/tomcat` and, depending on the operating system, add the following using the system environment `insserv`, `update-rc.d` or `chkconfig` command:

```
#!/usr/bin/bash

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

TCUSER=tomcat
TCDIR=/opt/tomcat

case "$1" in

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

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

restart)
    $0 stop
    $0 start

```



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

In FirstSpirit ServerManager under server properties, a "Tomcat" web server needs to be added using "PreviewTomcat" as the name, for example (see Chapter 7.3.12.5 page 276).

Parameters:

- Web server URL: leave blank
- Web directory: leave blank
- Tomcat user: see "username" under the "manager-script" role in the file `tomcat/conf/tomcat-users.xml`
- Tomcat password: the password associated with the user previously entered, look in the file `tomcat/conf/tomcat-users.xml`
- Tomcat Manager URLs:

Tomcat 7: <http://tomcathost:8080/manager/text>

Tomcat 6: <http://tomcathost:8080/manager/html>

(Local host name of the Tomcat server, as it is accessed from FirstSpirit Server, as well as Tomcat HTTP Port. If multiple Tomcat instances are set, the URLs are entered with a comma separating them.)

For supplementary information please refer to <https://community.e-spirit.com/docs/DOC-1772>.

Now in the server properties, switch from the FirstSpirit web application InternalJetty to PreviewTomcat and click "Install" to install the respective web applications on the Tomcat server. This installation takes place automatically using the script previously registered. Currently (in FirstSpirit 5.1), installation will need to be repeated for every FirstSpirit update in order to update the web applications.

If local project web applications are used (see Chapter 7.4.18 page 343), they will also have to be switched to PreviewTomcat.





If the `fs-server.jar` file has been updated, the FirstSpirit web applications installed on the Tomcat server will have to be reinstalled and reactivated on the server. This step must be performed in all projects for all web applications installed on the Tomcat web server. Updating is no automatic (see also 7.4.18.4 page 346).

4.5.5 External servlet engine and load balancing on multiple servlet engines

The load balancing configuration on two servlet engines is described in the FirstSpirit Community, see <https://community.e-spirit.com/docs/DOC-1781>.



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 Chapter 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/firstspirit5/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 ServerManager 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 Chapter 7.3.12 page 271). At first, add the desired external web server (see 7.3.12.3 page 275). 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 Chapter 7.3.13.5 page 282.
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 ServerManager as described in Chapter 7.3.13.5 (page 282 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. ServerMonitoring) can now be installed on the web server (see Chapter 7.3.13.5 page 282).

4.6.1 Integration into WebSphere Application Server

FirstSpirit does not have any requirements (exceeding those of the *Technical Data Sheet*) with regard to using this application server software. However, in practice, it is important to take note of the following anomalies, specifically when using IBM WebSphere.

In addition to the general configuration mentioned previously for integrating an external application server, the following configurations are necessary when using IBM WebSphere as the application server:

The file `firstspirit5/data/fslib/fs-webrt.jar` must be entered in the global class path of the WebSphere server, since the JARs digitally signed by e-Spirit cannot be loaded into the web applications under `WEB-INF/lib`. The file `fs-webrt.jar` should not be integrated as a "common library" file, but must instead be passed to WebSphere as a parameter of the Java VM.

The Java VM parameters are located in the web interface for WebSphere administration under "Server > Application Server > Server Name > Java and Process Management > Process Definition > Java Virtual Machine". There under "Class path", the complete path to the `fs-webrt.jar` file is specified; for instance, `/opt/firstspirit5/data/fslib/fs-webrt.jar`. If FirstSpirit Server is running on a server other than WebSphere, it is recommended that the directory `opt/firstspirit5/data/fslib` be passed via NFS to WebSphere so that the file `fs-webrt.jar` does not have to be copied manually to the WebSphere server every time FirstSpirit Server is updated.

Likewise, all JAR files that are used in modules delivered by e-Spirit need to be entered into the Java VM class path of the WebSphere server. The following procedure is necessary after activating a module for the Preview (`fs5preview`), ContentCreator (`fs5webedit`) or Staging (`fs5staging`) areas: first, open the WAR file(s) that were downloaded via the server and project properties using a ZIP program or `jar xvf fs_<Modulename>.war` and copy all JAR files contained within that with a naming convention of `fs-*.jar` from the `WEB-INF/lib` directory to the class path directory of the WebSphere server, where `fs-webrt.jar` has already been stored. In the WebSphere administration interface, add the copied JAR files to the class path, as was done for `fs-webrt.jar`, and restart WebSphere. This procedure is required for every FirstSpirit update.



In addition, the following entries are needed for "generic JVM arguments":
`-Djava.awt.headless=true -Dclient.encoding.override=UTF-8`

The following entry must be made in the `fs-server.conf` file for operation under WebSphere:

```
WEBAPP_ROOT_URL=/fs5root
```

(For more information on this parameter, see Chapter 4.3.1.7 page 46)

The FirstSpirit start page is then accessible on the WebSphere server as `http://fs5host.domain.net/fs5root`. To enable accessibility using the simpler URL `http://fs5host.domain.net`, the following redirect rule must be defined in the HTTP server of the WebSphere server:

```
RewriteEngine On  
RewriteRule ^/$ /fs5root/ [redirect,last]
```

In the `fs-server.conf` file, the parameter `preview.cacheFileWithTimestamp=*` must also be set (for more information on this parameter, see Chapter 4.3.1.8 page 49).



The WebSphere Application Server Java VM must be configured to regularly remove Java classes automatically for JSP files that no longer exist. In the case of Oracle Java VM, this is done by using the parameter

-XX:+CMSClassUnloadingEnabled

in conjunction with

-XX:+UseConcMarkSweepGC

To enable these changes in WebSphere, the WebSphere server needs to be restarted.



4.6.2 Logging for FirstSpirit web applications

Logging for FirstSpirit web applications when using external application servers is handled in some files on the (remote) FirstSpirit server in the directory `firstspirit5/log`, whose name is based on the format

```
fs-webapp-[Hostname]-[Port].log
```

`fs-webapp-localhost-1088.log`, for instance. The host name and port number refer to the application server on which the web application is installed. By default, `WARN` and higher level messages are copied to the log file. For debugging purposes, however, the level can be set lower via the parameter `WEBAPP_LOG_LEVEL` in the `fs-server.conf` configuration file (for more information, see Chapter 4.3.1.7 page 46).

The transfer is not in real-time; duplicates are filtered out. As in the case of other FirstSpirit log files, the `fs-webapp` log files are also compressed and archived as `*.gz` files. These include a time stamp in the file name. Archived files can be deleted or moved to `firstspirit5/backup` using the "Clean up logs" server schedule entry.

4.6.3 External application server requirements

The FirstSpirit web application requirements correspond first and foremost to those of any web application: uninterrupted operation is ensured. This means above all that it is important to plan ahead when configuring the Java VM heap size and garbage collector settings. To configure the Java VM of an external application server, the same basic principles apply as described in Chapter 4.3.2.1 "Configuration of Java VM" with regard to the FirstSpirit server. The following parameters are used as a configuration reference point if Java VM from Oracle 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 `-Xms` parameter defines the initial heap size (corresponds to `wrapper.java.initmemory` in `fs-wrapper.conf`), the `-Xmx` parameter defines the maximum heap size (corresponds to `wrapper.java.maxmemory` in `fs-wrapper.conf`). A maximum value of 75% of the available



main memory is recommended. The heap size selected should not be too large, since FirstSpirit attempts to fill the entire free space for cache. The parameter `com.sun.management.jmxremote` is used to monitor the heap capacity, even in production mode, so that trends related to the allocation of storage can be detected and the heap size can be increased, if necessary. Current monitoring systems offer a JMX interface that allows monitoring via web application servers. The "Lambda Probe" web application is recommended for querying the current values (<http://www.lambdaprobe.org>). In addition, the `jconsole` program (<http://docs.oracle.com/javase/1.5.0/docs/guide/management/jconsole.html>) can be used for interactive queries. It is a component of every JDK (not JRE) by Oracle.

Particularly in the case of application servers, the configuration of the parameter `MaxPermSize` is essential, since a Java class is created in this memory area for each JSP file used. The area selected must also be sufficient in size. JMX can be used to query the current allocation.



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 (ContentCreator, start page and ServerMonitoring) 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 Chapter 4.7.2.2 page 145). Generate an independent test certificate with a different host name via the following call:

```
keytool -genkeypair -alias fs5.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 (Chapter 4.7.2.2 page 145).

In order to enable use of the self-signed test certificate on sites of the FirstSpirit SiteArchitect, 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 SiteArchitect 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`²¹ 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 fs5.yourdomain.net -keyalg RSA -keysize 2048 -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. `fs5.yourdomain.net`.

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

```
keytool -certreq -keystore conf/fs-keystore.jks -storepass changeit -alias fs5.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 fs5.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”:

²¹ <http://www.openssl.org/>



```
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 2048
```

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

The HTTPS listener must be enabled in configuration file `fs-webapp.xml`. The configuration parameters are already entered in the file but are disabled using comment characters "`<!-- -->`". The following configuration parameters are necessary (case-sensitive):

```
<!--
HTTPS-Connector
=====
* for additional parameters read http://wiki.eclipse.org/Jetty/Howto/Configure_SSL
* if NIO is not available, use org.eclipse.jetty.server.ssl.SslSocketConnector

-->
<New id="sslContextFactory" class="org.eclipse.jetty.http.ssl.SslContextFactory">
<Set name="keyStore"><SystemProperty name="cmsroot" />/conf/fs-keystore.p12</Set>
<Set name="keyStorePassword">PASSWORD</Set>
<Set name="keyStoreType">pkcs12</Set>
</New>
<Call name="addConnector">
<Arg><New class="org.eclipse.jetty.server.ssl.SslSelectChannelConnector">
<Arg><Ref id="sslContextFactory"/></Arg>
<Set name="Port">8443</Set>
<Set name="maxIdleTime">30000</Set>
<Set name="Acceptors">2</Set>
<Set name="AcceptQueueSize">100</Set>
</New></Arg>
</Call>
```

"Password" refers to the password for the keystore and "KeyPassword" refers to the password for the certificate (or private key), which is normally the same as the one for the keystore.

If a "JKS" type keystore is to be used, "pkcs12" must be replaced by "jks".

With Windows, the file extension `.pfx` is used instead of `.p12` for pkcs12 files.



If the HTTPS certificate requires intermediate certificates, as is now generally the case with all certificate authorities, and these still need to be added to the keystore, we recommend the following procedure:

- In the example, the private key and the received server certificate, without the intermediate certificates, are in file `servercert.pfx` in the format PKCS12, which is the usual format for Java-based web servers.
- First of all, the private key (including the server certificate) must be converted to PEM format:

```
openssl pkcs12 -export -in servercert.pfx -out serverkeycert.pem
```
- Now use a text editor to open the `serverkeycert.pem` file and copy the received private key and the certificate into the corresponding individual files `serverkey.pem` and `servercert.pem`. Transfer the content between lines "---- BEGIN..." and "---- END..." and the lines themselves, and remove the rest.
- In the example, the intermediate certificates are in the `intermediate1.pem` and `intermediate2.pem` files, where the certificate chain is as follows: Certificate authority (CA) -> Intermediate 1 -> Intermediate 2 -> Server certificate.
- If the individual certificates are in DER format, e.g., with the file extension `.cer` or `.crt`, i.e., they do not contain ASCII text with "---- BEGIN..." lines that can be read directly, the individual files must first be converted to PEM format:

```
openssl x509 -in intermediate1.cer -inform DER -out intermediate1.pem -outform PEM
```

```
openssl x509 -in intermediate2.cer -inform DER -out intermediate2.pem -outform PEM
```
- All of the certificates are now incorporated in one file and **must follow the sequence** specified in the certificate chain, which starts with the certificate authority and ends with the server certificate. If the sequence is altered, the server does not display an error message on startup and a certificate error is only shown when a connection is established via Java Web Start.

```
cat intermediate1.pem intermediate2.pem servercert.pem > serverall.pem
```
- Finally, the private key and the certificate chain are incorporated in a single PKCS12 file; enter the password that matches that of the `fs-webapp.xml` file:

```
openssl pkcs12 -export -inkey serverkey.pem -in serverall.pem -out fs-keystore.p12 -passout pass:PASSWORD
```



- The following temporary files should be removed:
servercert.pem
serverkey.pem
serverkeycert.pem
serverall.pem
intermediate1.pem
intermediate2.pem
- As a final step, move the fs-keystore.p12 file to the firstspirit5/conf folder.

After restarting the FirstSpirit Server, the start page is now also available via <https://fs5.yourdomain.net:8443> next to <http://fs5.yourdomain.net:8000>.

You can find further information on the HTTPS configuration of the Jetty web server used by FirstSpirit at:

http://wiki.eclipse.org/Jetty/Howto/Configure_SSL

4.8 Additional security measures (FirstSpirit 5.1R4 and higher)

4.8.1 Parameterizing encryption

Another component of the FirstSpirit security concept is the ability to encrypt internal communication. A parameter in the `fs-server.conf` file can be used to instruct the FirstSpirit Server to only accept connections with a predefined encryption type (e.g. TLS) (`ALLOWED_ENCRYPTIONS=1`, see Chapter 4.3.1.1, page 33).

This optional encryption function can be configured for all internal communication between the FirstSpirit server, FirstSpirit cluster nodes, FirstSpirit SiteArchitect, and the FirstSpirit web applications. Primarily, this includes:

- Server-server communication
(e.g. front-end server – FirstSpirit Server, generation server – FirstSpirit Server)
- Client-server communication
(e.g. SiteArchitect – front-end server (HTTPs)/FirstSpirit Server (socket), Web browser – front-end server)

In FirstSpirit Version 5.1R4 and higher, advanced configuration options are available for encryption. It is now possible to define precisely which protocol version and which algorithms should be used for the encryption process. In addition, certificates for server and client



authentication (two-way authentication) are supported. On the client side, any cluster nodes that are present must be taken into account in addition to the FirstSpirit web applications.

On the server side, configuration relies on the `fs-server.conf` file (including for cluster nodes). On the client side, it relies on Java or system parameters on the application server or utilizes the `web.xml` file of the web applications. In the case of FirstSpirit SiteArchitect, encryption is configured via `-D` parameters in the connection settings.

As there can be no assurance that all FirstSpirit installations will feature a certificate store, the default configuration settings only allow for conventional encryption (without authentication and certificates). These security settings must be configured separately by the server administrator:

- Configuration of the FirstSpirit server (see Chapter 4.3.1.20 page 70)
- Configuration of the web applications and servlets (see Chapter 4.8.1.1, page 149)
- Configuration of the cluster nodes (see Chapter 4.8.1.2, page 151)
- Configuration of FirstSpirit SiteArchitect (see Chapter 4.8.1.3, page 152)

For a full description of the configuration parameters, see 4.3.1.20 page 70. The parameters are written differently depending on whether they are stored in the `fs-server.conf` file or, for example, as system parameters on the application server. However, exactly the same functional description applies in both cases.

If certificates are to be used for authentication, valid certificates must first be created for the server and, where applicable, for the clients as well. The necessary steps are described in Chapter 4.7.2 (page 144).

4.8.1.1 Configuring encryption in web applications and servlets

For a detailed description of the encryption parameters, see Chapter 4.3.1.20 (page 70 ff).

The encryption parameters must be transferred to the configuration for the relevant web applications and servlets. This can be achieved in various ways.

Via the system environment: Depending on the operating system, encryption can be configured directly on the application server using parameters that are specific to the operating system.

In a Windows environment (for example) via:

```
set FIRSTSPIRIT_ENCRYPTION=1
set FS_SSL_PROTOCOLS=TLSv1.2
set FS_SSL_CIPHER_SUITES=DEFAULT
```



```
set FS_SSL_NEED_CLIENT_AUTH=TRUE
set FS_SSL_KEY_STORE=/HOME/SERVER_CERT.JKS
set FS_SSL_KEY_STORE_PASSWORD=q1w2e3r4t
```

In a Linux environment (for example) via:

```
export FIRSTSPIRIT_ENCRYPTION=1
export FS_SSL_PROTOCOLS=TLSv1.2
export FS_SSL_CIPHER_SUITES=DEFAULT
export FS_SSL_NEED_CLIENT_AUTH=TRUE
export FS_SSL_KEY_STORE=/HOME/SERVER_CERT.JKS
export FS_SSL_KEY_STORE_PASSWORD=q1w2e3r4t
```

Via the Java environment: as a -D Java property, e.g. using:

```
-Dfirstspirit.encryption=1
-Dfs.ssl.protocols=TLSv1.2
-Dfs.ssl.cipherSuites=DEFAULT
-Dfs.ssl.needClientAuth=true
-Dfs.ssl.keyStore=/home/server_cert.jks
-Dfs.ssl.keyStorePassword=q1w2e3r4t
```

Via the servlet configuration: For specific applications, encryption can also be configured using servlet context parameters (in web.xml), e.g.:

```
<context-param>
  <param-name>firstspirit.encryption</param-name>
  <param-value>1</param-value>
</context-param>
<context-param>
  <param-name>fs.ssl.protocols</param-name>
  <param-value>TLSv1.2</param-value>
</context-param>
<context-param>
  <param-name>fs.ssl.cipherSuites</param-name>
  <param-value>DEFAULT</param-value>
</context-param>
```

Evaluation order: The following evaluation order applies here (from highest priority to lowest priority):



1. Configuration of the parameter via the Java environment
2. Configuration of the parameter via the system environment
3. Configuration of the parameter via the `web.xml` file

As an alternative to the FirstSpirit keystore parameters, the Java keystore parameters can be used instead (see Chapter 4.8.1, page 148).

4.8.1.2 Configuring encryption for cluster nodes

Via the `fs-server.conf` file (for cluster nodes only): In the case of cluster nodes, the encryption parameters can either be configured centrally using the `fs-server.conf` file (of the master) or the `fs-wrapper.slave.conf` file (see below).

Specific cluster nodes can be individually configured by adding the prefix `cluster.<NODE>.<PROPERTY>`, e.g.:

```
cluster.slave1.firstspirit.password=uNuegFThpxtvD23C
cluster.slave2.firstspirit.password=KhPXSNBuoJzhWZ1M
```

All cluster nodes can be universally configured by adding the prefix `cluster.<PROPERTY>`, e.g.:

```
cluster.firstspirit.password=uNuegFThpxtvD23C
```

Example of how to configure the encryption parameters in `fs-server.conf`:

```
fs.ssl.protocols=TLSv1.2
fs.ssl.cipherSuites=DEFAULT
fs.ssl.needClientAuth=true
fs.ssl.keyStore=/home/server_cert.jks
fs.ssl.keyStorePassword=q1w2e3r4t

#cluster configuration for all cluster nodes
cluster.firstspirit.password=<globalPW>
cluster.firstspirit.encryption=1
cluster.fs.ssl.cipherSuites=DEFAULT
cluster.fs.ssl.keyStore=/home/user/selfsigned.jks
cluster.fs.ssl.keyStorePassword=OBF:changeit123

#cluster configuration for individual nodes e.g.:
cluster.slave1.firstspirit.password=1234
```



```
cluster.slave2.firstspirit.password=5678
```

Via the `fs-wrapper.slave.conf` file (for cluster nodes only): In the case of cluster nodes, the encryption parameters can also be configured using Java properties in the `fs-wrapper.slave.conf` file, e.g.:

```
wrapper.java.additional.5=-Dfirstspirit.encryption=1.
```

Evaluation order: The following evaluation order applies here (from highest priority to lowest priority):

1. Configuration of the parameter using the `cluster.<NODE>` prefix (individual configuration for a cluster node) and the `fs-server.conf` file
2. Configuration of the parameter using the `cluster.` prefix (universal configuration for all cluster nodes) and the `fs-server.conf` file
3. Configuration of the parameter using -D properties in the `fs-wrapper.slave.conf` file

4.8.1.3 Configuring encryption for FirstSpirit SiteArchitect

The encryption parameters for SiteArchitect are configured as -D properties using the Web Start connection settings on the client side (see Chapter 7.3.7, page 260). Configuration is not possible on the server side because the parameters are required to establish the connection successfully and so cannot be loaded from the server.

If the FirstSpirit Server uses a valid certificate (excerpt from the `fs-server.conf` file):

```
fs.ssl.cipherSuites=DEFAULT
fs.ssl.keyStore=/home/server_cert.jks
fs.ssl.keyStorePassword=q1w2e3r4t
```

the following parameter must be specified in the connection settings for starting SiteArchitect:

```
-Dfs.ssl.cipherSuites=DEFAULT
```

If a valid client certificate also has to be used because the `fs.ssl.needClientAuth=true` parameter has been configured in the `fs-server.conf` file, the following parameters must be specified in the connection settings for starting SiteArchitect:

```
-Dfs.ssl.cipherSuites=DEFAULT
-Dfs.ssl.keyStore=/home/user/client_cert.jks
-Dfs.ssl.keyStorePassword=OBF:geheim123
```



If the SSL handshake does not work with these settings, the integrated logging feature in Java can be enabled by specifying the following parameter (client and server respectively):

```
-Djavax.net.debug=ssl
```

4.8.2 Repository encryption

FirstSpirit uses repositories to maintain version histories of project data. The repository is a central location for managing the file structures required by the content management system (media, pages, templates, etc.). There is a separate repository for each project. Data is written to the repository whenever an action is performed in FirstSpirit (e.g. when elements are created, edited or deleted).

In FirstSpirit Version 5.1R4 and higher, repository files (structures, content, media) can be saved in an encrypted format. This involves performing the following steps:

1. Create a global key file for the FirstSpirit Server
(see Chapter 4.8.2.1, page 153)
2. Configure encryption on the server – the minimum requirement for encryption is that the `repository.encryption.keyFilePath` parameter must be configured (see Chapter 4.3.1.11, page 56)
3. Configure project-specific encryption
(see Chapter 7.4.22, page 366)
4. Perform encryption/decryption
(see Chapter 4.8.2.2, page 154)

The actual encryption process is handled by the Java Cryptography Extension (JCE). All symmetric encryptions and modes supported by the relevant Java platform are possible. This depends on which Java version is used and whether "Unlimited Strength Jurisdiction Policy Files" (see <http://www.oracle.com/technetwork/java/javase/downloads/jce-7-download-432124.html>) are installed.

4.8.2.1 Creating the key file

The first step is to create a key file using a key of your choice. This global server key must be at least eight bytes long. The content of the specified file must be encoded in UTF-8. White spaces at the beginning and end of the file are ignored.

The file must be saved in a suitable location.





Access to the global server key file should be properly secured to prevent unauthorized persons from accessing the repository contents. At the same time, this means that if the key file is damaged or lost, it will no longer be possible to access the contents of the repository.

The path to the key file must be stored in `fs-server.conf` (for information on configuration, see Chapter 4.3.1.11, page 56). Only then can encryption of the FirstSpirit project repositories be enabled in the project settings.

4.8.2.2 Performing encryption/decryption

If you enable the "Encryption enabled" option for the project (see Chapter 7.4.22, page 366) and confirm your choice by pressing "OK", encryption of the data commences in accordance with the desired settings. The relevant project is deactivated during the process. Given that encryption can take a little while, it should only be performed during a maintenance period.

If you deactivate the "Encryption enabled" option and confirm your choice by pressing "OK", the data for the relevant project is decrypted using a similar process to the one described above.



The encryption/decryption process must not be interrupted because this can result in an undefined project state.

If changes are required (e.g. due to a system failure during encryption), please contact <https://helpdesk.e-spirit.com>.



4.9 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. By means of this architecture 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.

A project export or import usually occurs from one database system to another.

4.9.1 Storing the JDBC driver files

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

1. As a FirstSpirit module: (recommended)

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 Descriptor file 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 4.9.4.2 page 171).

2. Via the `shared/lib` directory:

The relevant drive files must be copied into the `firstspirit5/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.



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

4.9.2 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 4.9.2.1 Page 156)
- file `module.xml` (see Chapter 4.9.2.2 Page 156)

optionally:

- file `web.xml` (see Chapter 4.9.2.2 Page 156)

These files must be integrated by means of an FSM file (ZIP archive) to a FirstSpirit module (see Chapter 4.9.2.3 Page 161).

4.9.2.1 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 restrictions and notes about the drivers of the specific database types see also Chapter 4.9.6 page 177 and the other sub chapters of Chapter 4.9.

4.9.2.2 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 9.1** database:

```
<module>
  <name>JDBC_PostgreSQL_9_1</name>
  <displayname>Database Driver PostgreSQL 9.1</displayname>
  <version>9.1.902.1</version>
  <description>JDBC Driver for PostgreSQL 9.1 databases</description>
  <vendor>PostgreSQL Global Development Group</vendor>

  <resources>
    <resource scope="module">lib/postgresql-9.1-902.jdbc4.jar</resource>
  </resources>

  <components>
    <web-app>
      <name>WebApp_PostgreSQL_9_1</name>
      <displayname>WebApp PostgreSQL 9.1</displayname>
      <description>Provides the JDBC Driver in a web
application.</description>
      <web-xml>web.xml</web-xml>
      <web-resources>
        <resource scope="module"
          name="postgresql"
          version="9.1.902"
          minVersion="9.1.1"
          maxVersion="9.1.9999">lib/postgresql-9.1-902.jdbc4.jar
        </resource>
      </web-resources>
    </web-app>
  </components>

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

The module consists of two parts: one part defines the resources for the FirstSpirit Server, the other part for the web applications (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 technical name for the components. Only the following characters are allowed: capital and lower case letters (A-Z, a-z) and figures (0-9). This name will be validated when installing the module. Modules which do not comply with this convention can not be installed.



This technical name will be used for example for the display in the FirstSpirit ServerManager (in case where no display name was defined for the module), for checking the module when updating and installing and for creating files and folders on the hard disk. The name which is assigned to the server component must be indicated in the database layer configuration, too (see Chapter 4.9.3.2 page 163).

`<displayname>`: This tag can be used for assigning an optional display name for the module.

If a display name is defined this will be shown in all FirstSpirit user interfaces, for example in the overview of modules of the FirstSpirit server (see Figure 4-7). The display name which is assigned to the web application component will also be used in the Project properties, are "Web components" (see Figure 4-11). The mandatory attribute `<name>` will still be used as unique technical name. If no display name is defined the technical name will be shown in the user interfaces.

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

`<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. *9.1.902* for version 9.1 build 902.

`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 9.1 until 9.1.999 can be used. If a second driver is provided



by another module, e.g. build 903, this can be used also from 9.1 until 9.1.999. In this case only the higher driver version (i.e. 903) 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.

<DRIVER>: Contains the complete class name of the used JDBC driver, e.g. `org.postgresql.Driver` for PostgreSQL. (See also parameter `DRIVER` in Chapter 4.9.4.1 page 169 and the chapters containing the database specific sample configurations in Chapter 4.9.7 page 179.)

<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 4.9.4.1 page 169 and the chapters containing the database specific sample configurations in Chapter 4.9.7 page 179.)

If the JDBC driver should be available in a web application, the file `web.xml` is required (see Chapter 4.9.2.2 page 156):

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="JDBC_PostgreSQL_9_1"
  version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"/>
```

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:

```
<module>
  <name>JDBC_Derby_10</name>
  <version>10.8.2.2.1</version>
  <description>JDBC Driver for Derby 10.8 databases</description>
  <vendor>Apache Software Foundation</vendor>

  <resources>
    <resource scope="module">lib/derbyclient.jar</resource>
  </resources>

  <components>
    <web-app>
      <name>WebApp_Derby_10</name>
      <description>Provides the JDBC Driver in a web
application.</description>
      <web-xml>web.xml</web-xml>
      <web-resources>
        <resource scope="module"
          name="derby"
          version="10.8.2.2"
          minVersion="10.8.1"
          maxVersion="10.8.9999">lib/derbyclient.jar</resource>
      </web-resources>
    </web-app>
  </components>

  <configuration>
    <DRIVER>org.apache.derby.jdbc.ClientDriver</DRIVER>
    <layerclass>de.espirit.or.impl.derby.DerbyLayer</layerclass>
  </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="JDBC_Derby_10"
  version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"/>
```

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 4.9.3.4.1



Page 166).



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).

4.9.2.3 Directory structure of a JDBC driver module

If the driver is only used for the FirstSpirit Server, the files must be deposited in the following directory structure (cf. Chapter 4.9.2.1 page 156 and Chapter 4.9.2.2 page 156):

[-] jdbc_postgresql_9_1		
lib		
postgresql-9.1-902.jdbc4.jar	Executable Jar File	549 KB
META-INF		
module.xml	XML-Dokument	1 KB

Figure 4-5: Directory structure server

If it should be used also in web applications the file `web.xml` must be integrated on the highest level:

[-] jdbc_postgresql_9_1		
web.xml	XML-Dokument	1 KB
lib		
postgresql-9.1-902.jdbc4.jar	Executable Jar File	549 KB
META-INF		
module.xml	XML-Dokument	1 KB

Figure 4-6: Directory structure server and web application

To get a valid FirstSpirit module, a ZIP file must be created from the content of the superordinate folder ("jdbc_postgresql_9_1"). 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 `jdbc_postgresql_9_1.fsm`.

As an alternative, the module file can be created using the following command:

```
jar cvf jdbc_postgresql_9_1.fsm -C jdbc_postgresql_9_1 .
```



The programme "jar" is part of each JDK and can be found, depending on the operating system and installation, for example under `c:\programmes\jdk[versionnumber]\bin\jar.exe` or `/opt/jdk[versionnumber]/bin/jar`.

4.9.3 Installation and configuration of the JDBC driver module

4.9.3.1 Installation of the JDBC driver module

If the JDBC driver module has been created successfully as described in Chapter 4.9.2 page 156, it must be installed on the FirstSpirit Server. This will be carried out by means of the ServerManager.

For this purpose, the button "Install" must be clicked in the Server properties in the area "Modules" (see Chapter 7.3.11 page 265). 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: `JDBC_PostgreSQL_9_1`, see Chapter 4.9.2.2 page 156, tag `<name>`) and version (tag `<version>`):

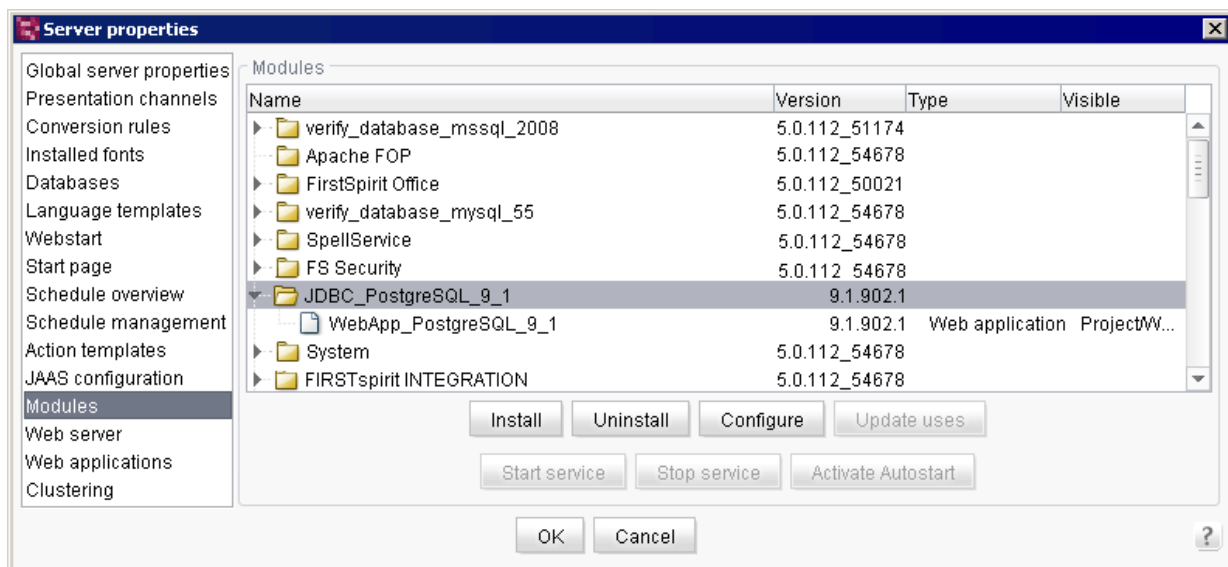


Figure 4-7: Server properties – JDBC driver as module

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.



4.9.3.2 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" (see also Chapter 7.3.5 page 252):

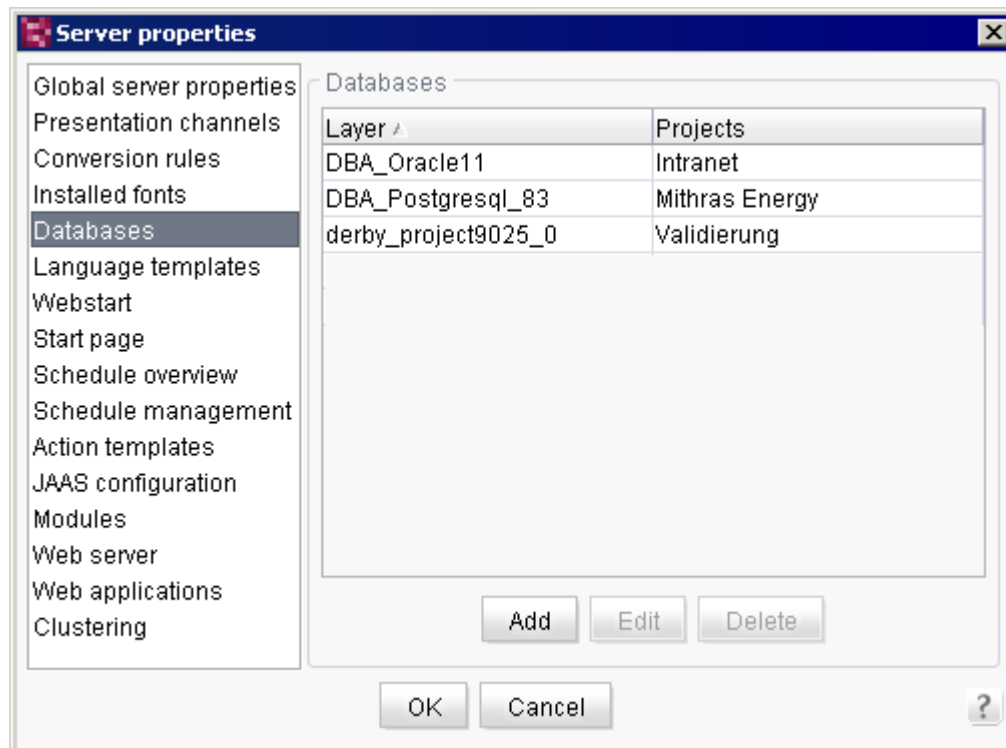
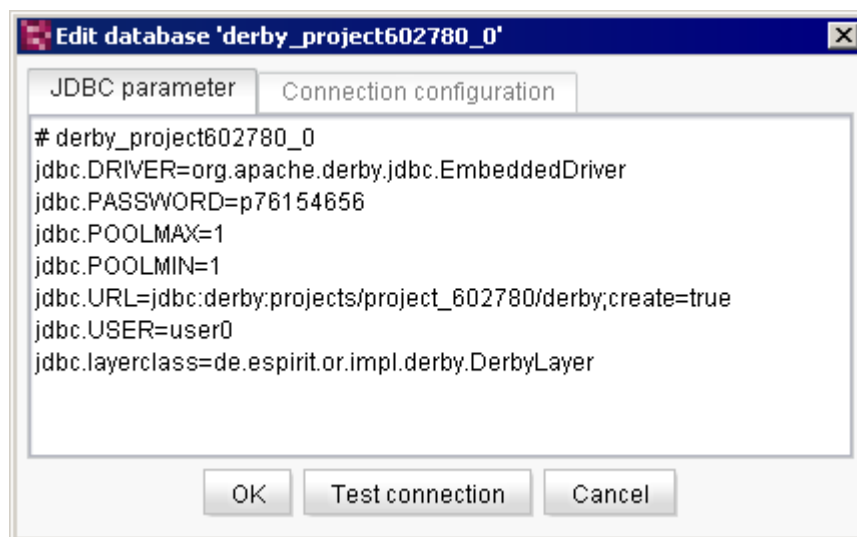


Figure 4-8: Server properties – Databases

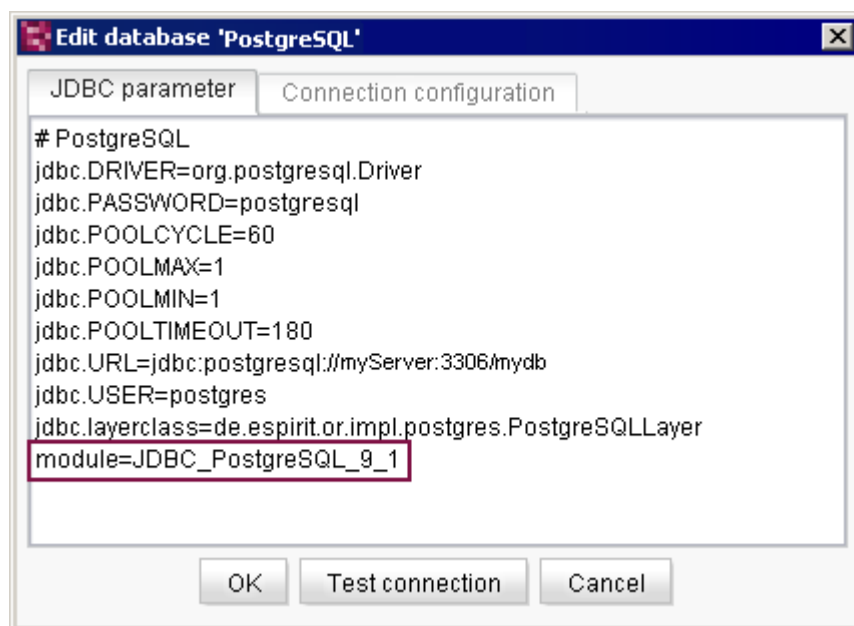
Double-click on the respective entry to open the dialog "Edit database". Here, you can edit the JDBC parameters for the database connection:



**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=JDBC_PostgreSQL_9_1
```

**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".



4.9.3.3 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" (see Chapter 7.4.18 page 343). 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 4.9.2.2 page 156):

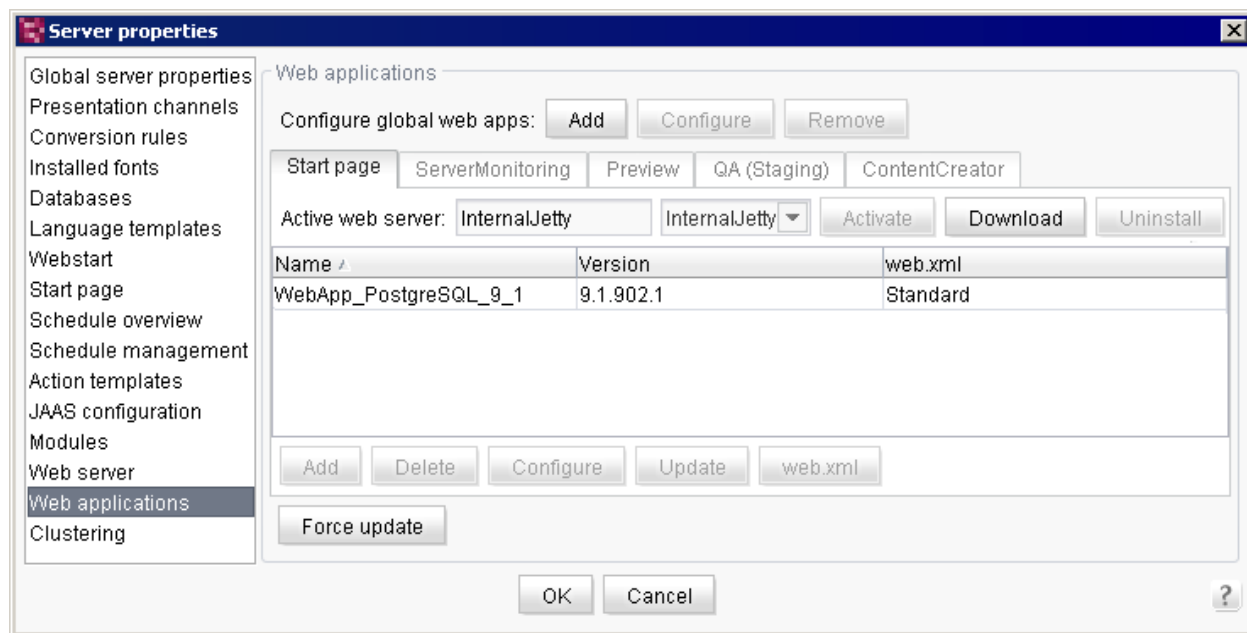


Figure 4-11: Project properties – JDBC driver as web component

The component of the JDBC driver module can not be configured furthermore.



4.9.3.4 Usage of the Derby database in web applications

If a Derby database is used in web applications (e.g. in the FirstSpirit module `DynamicDatabaseAccess`), the JDBC driver module must be added to the web application as well, as described in Chapter 4.9.3.3 page 165.



The Derby database, integrated in FirstSpirit, is not dedicated for productive operation and should be used for test purposes only.

4.9.3.4.1 Example: Module "FirstSpirit DynamicDatabaseAccess"

When using the module "FirstSpirit DynamicDatabaseAccess" 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 4.9.3.2 page 163) 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 "DynamicDatabaseAccess" 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:



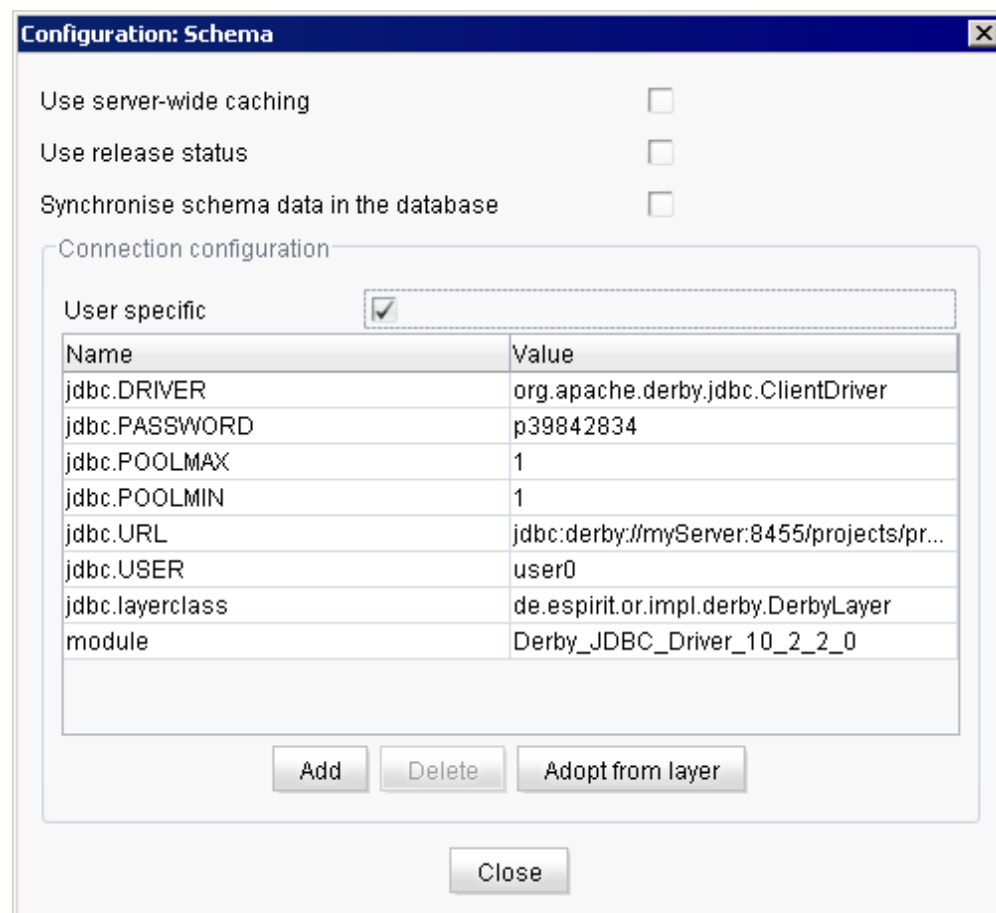


Figure 4-12: FirstSpirit DynamicDatabaseAccess – Database configuration

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 4.9.3.2 page 163).

4.9.3.4.2 Individual implementations of modules

You must carry out these adjustments, which are carried out for the example "FirstSpirit DynamicDatabaseAccess" (see Chapter 4.9.3.2 page 163), in each module which has been developed individually and which works together with the Derby database.



4.9.4 Data source configuration

The database integration of the FirstSpirit Server is configured via the ServerManager (see Chapter 7.3.5 page 252) or directly via “DATABASES” in configuration file `fs-database.conf` (see Chapter 4.3.3 page 84). It is recommended to use the ServerManager (see Chapter 7.3.5 page 252) 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 7-38).

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 configuration via the ServerManager (recommended) or directly via the file `firstspirit5/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 Chapter 4.9.4.1 page 169.

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 Chapter 4.9.4.2 page 171.

4.9.4.1 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 Chapter 4.9.1). 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. The integration of JDBC driver files as FirstSpirit module is recommended (see also Chapter 4.9.2 page 156).

```
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 Chapter 4.9.7 page 179).

`<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 7.3.5 page 252).

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.



4.9.4.2 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 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 Online documentation*.

Example:

```
database.jdbc.SCHEMA=goodsdatabase
```



<database>.jdbc.POOLMAX: FirstSpirit uses one separate pool instance for every usage, e.g. one schema node in a project, one deployed web application (e.g. FirstSpirit DynamicDatabaseAccess) 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
```

`<database>.jdbc.isolation`: Transactions running simultaneously that make changes to data in the database could result in undefined states. Transaction isolation should be configured to prevent a running transaction from being changed to an undefined state by another transaction running at the same time (due to a change in the data



used). Different isolation levels²² can be configured using this parameter. The following are generally supported:

READ_COMMITTED: lowest isolation level.

REPEATABLE_READ: medium isolation level.

SERIALIZABLE: highest isolation level (default value).

Note: Not every database management system supports all isolation levels. For instance, OracleDB 11 supports only the READ_COMMITTED and SERIALIZABLE isolation levels, but not REPEATABLE_READ. In certain cases, you should consult the documentation of the relevant database for more information.

```
database.jdbc.isolation=READ_COMMITTED
```

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 4.9.2 page 156.

```
database.module=JDBC_PostgreSQL_9_1
```

4.9.4.3 Description of the Oracle-specific parameters

<database>.jdbc.oracle.TABLESPACE: Only relevant for DBA layer configurations. A tablespace must be specified via this parameter (required parameter), e.g.

```
jdbc.oracle.TABLESPACE=USERS
```

4.9.4.4 Description of the MS-SQL specific parameters

<database>.jdbc.CATALOG: The meta data of the databases 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 ORMMapper works with.

```
database.jdbc.CATALOG=ormapper
```

²² For a description, refer to the ANSI standard or documentation on the database to be used.



4.9.5 Required permissions for database user accounts

Depending on the used database specific permissions are necessary. These are described in the following Chapters.

4.9.5.1 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>;
```

4.9.5.2 MySQL databases

For DBA layers:

```
# mysqladmin --default-character-set=utf8 create <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 <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
```



4.9.5.3 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
```

4.9.5.4 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



4.9.6 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 4.9.7 from page 179.

4.9.6.1 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.
5. The number of columns should be kept as low as possible.
6. References to entries in external databases

Note about reference graph support for database content: the reference graph of a project is a central part of many FirstSpirit functions (e.g. "Show usages"). References to entries in external databases can only be added to the reference graph if numeric primary keys are used.

4.9.6.2 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 too 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 than 320 characters) or
- 1 language with 11 string columns (each more than 730 characters)

4.9.6.3 Oracle



When using Oracle databases, the saving of database schemes and changes to them can take some time.

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.



4.9.6.4 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 SiteArchitect, 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
```

4.9.7 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.

4.9.7.1 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>.



4.9.7.2 Configuration example: MS-SQL-Server

Driver: sqljdbc-x.x.jar

```
jdbc.CATALOG=testDB
jdbc.DRIVER=com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbc.PASSWORD=testpassword
jdbc.URL=jdbc:microsoft:sqlserver://myserver:1433;DATABASENAME=testDB;select
Method=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=testpassword
jdbc.URL=jdbc:microsoft:sqlserver://myserver:1433;DATABASENAME=testDB;select
Method=cursor
jdbc.USER=testuser
jdbc.layerclass=de.espirit.or.impl.mssql.MSSQL2005Layer
```

For description of the MS-SQL specific parameters see chapter 4.9.4.4 page 174.

4.9.7.3 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`.



4.9.7.4 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.

4.9.7.5 Configuration example: DB2

Layer parameter:

For configuring an IBM DB2 data base a JDBC type 4 driver must be used:

```
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
```

Driver: db2java.zip (must precisely fit the DB2 server used)

Port: DB2 Java connector (is provided via: `db2jstrt PORTNUMBER`).

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:

- since DB 9.5: `db2jcc4.jar`, `db2jcc_license_cu.jar`
- before DB 9.5: `db2jcc.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 DB2 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.

4.9.7.6 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 218) and write access to the database is set for this project (see chapter 7.4.12 page 325). A Derby database can be subsequently created in the server properties (see chapter 7.3.5 page 252) and added to the project properties of the project (see chapter 7.4.12 page 325).

In order to use the Derby database from external processes, e.g. web application with FirstSpirit module DynamicDatabaseAccess in the external application server, the JDBC connector must first be activated for network connections (see chapter 4.3.1.13 page 60). 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 "fs5server" and the Project ID 12345 is replaced by the actual ID:

```
jdbc.URL=jdbc:derby://fs5server: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 `firstspirit5/log/fs-database.log`. The Derby version used in FirstSpirit is now 10.8.2.2.

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 `firstspirit5/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://fs5server:1527/projects/project_12345/derby;DATABASENAME=P{SCHEMA-ID}_{PROJECT-ID}
```

It is recommended that JDBC drivers be integrated as a FirstSpirit module instead of in `firstspirit5/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 4.9.2 page 156.

4.9.8 Procedure for connecting external databases

1. ServerManager: Configure a new database connection in the **Server properties** (menu item “Server” / “Properties” / “Databases”; see Chapter 7.3.5 page 252). 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. ServerManager: 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 Chapter 7.4.12 page 325).

Databases				
Name	Selected	Read only	No schema sync	
derby_project36280_0	<input type="checkbox"/>			▲
derby_project41177_0	<input checked="" type="checkbox"/>			■
external_database	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	▼

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*).

4.10 Roll-out process for native applications

4.10.1 Roll-out process (server)

A Mozilla engine is used in FirstSpirit within the scope of the "Integrated preview" (inline preview) function. The native system components required for this must first be distributed to the users' workstation systems.

The `fs-server.jar` file contains for this purpose 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 `~FirstSpirit5\data\clientapp` 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 4.3.1.2 page 37).

Each client application is then located in an individual directory below `~FirstSpirit5\data\clientapp` (default setting), for example, the Mozilla engine in Windows operating systems under:

```
~FirstSpirit5\data\clientapp\jxbrowser\windows\lib
```

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, `clientapp\jxbrowser\windows` or `clientapp\jxbrowser\osx`.



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.

In order to make changes manually, e.g. to install plug-ins or to disable existing problematic plug-ins, the following steps must be taken:

1. Stop FirstSpirit Server
2. Delete the `version.txt` file (under
 `~FirstSpirit5\data\clientapp\jxbrowser`)
3. Make the desired changes below the directory `\jxbrowser`, e.g. add or remove files
 under `~FirstSpirit5\data\clientapp\jxbrowser\...\plugins`
4. Start FirstSpirit Server

The changes will take effect automatically the next time the SiteArchitect is started.

If the changes need to be removed at a later date, such as when the original version of the application that came with FirstSpirit is to be used again:

1. Stop FirstSpirit Server
2. Delete the directory `~FirstSpirit5\data\clientapp\jxbrowser`
3. Start FirstSpirit Server

The next time a SiteArchitect is started, the default version of the application will be used.

4.10.2 Roll-out process (workstation computer)

The directories created on the server on rolling out the native system components (see Chapter 4.10.1 page 184), are created as an exact copy in the User home directory of the users' 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

`\users\<username>\.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:\Users\<USERNAME>\.firstspirit_5.1R4\jxbrowser\xulrunner
```



If Internet Explorer is used, the client application executables are not rolled out to the directory referred to above. Instead, they are placed in a temporary directory under the user home directory, e.g.: C:\Users\<username>\AppData\Local\Temp\FirstSpirit_123456536456

This behavior (which is specific to Internet Explorer) cannot be prevented by FirstSpirit. If Internet Explorer is being used, you should ensure that the appropriate execution permissions have been configured for the temporary directory as well (see Chapter 4.10.5, page 188).

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 4.3.1.2 page 37 or Chapter 6.3.3.1 page 200 and Chapter 4.10.5 page 188). 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. See also Chapter 4.10.5 page 188.



4.10.3 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:

```
~FirstSpirit5\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:\Users\<USERNAME>\.firstspirit_5.1R4\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.

4.10.4 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.

4.10.5 Roll-out directory requirements

The following are required for the directory in which the browser integration components will be rolled out so that the integrated preview works correctly:

1. The user must have write permissions for the following directory:
`.../.firstspirit/jxbrowser/jxProfile`
2. The user must have the "Execute" permission for the following directory:
`.../.firstspirit/jxbrowser/lib`
`.../.firstspirit/jxbrowser/xulrunner`
3. The user home directory (the directory above `.firstspirit`), as well as the entire file path up to and including `.firstspirit` must not contain any special characters.
4. If Internet Explorer is used, execution permissions are also required for the temporary directory that is specific to the operating system, e.g.:
`.../AppData/Local/Temp/FirstSpirit_<Number>`
(for reasoning, see Chapter 4.10.2, page 185).

You could otherwise receive error messages, for instance,

`Could not intialize browser!`

and the integrated browser will not open.



5 FirstSpirit web application configuration



Logging onto multiple FirstSpirit servers simultaneously with the same host names (e.g. myServer:8200 and myServer:8400) via a web browser is not supported.

5.1 FirstSpirit start page configuration (fs5root)

The default FirstSpirit Server start page is located under:

```
<cms_basedir>\web\fs5root\index.jsp
```

A valid login is required for calling the start page (see chapter 6 page 194). The start page can be used to provide users with simple access to the FirstSpirit applications (see Chapter 6.3 page 196).

The FirstSpirit start page can be configured and adapted via the FirstSpirit ServerManager (see Chapter 7.3.7 page 260).

The quick start entries can also be configured via the FirstSpirit ServerManager (see Chapter 7.3.8 page 261).

There is also the possibility to configure the connection settings user-specifically (see Chapter 6.3.3.1 page 200).

5.2 ContentCreator configuration

ContentCreator has been developed as an extension to the editorial system FirstSpirit SiteArchitect. The ContentCreator 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 SiteArchitect installation, a Java environment (JRE) is not required for ContentCreator. Technically speaking, ContentCreator only works on the basis of HTML and JavaScript.





The ContentCreator configuration occurs among other things via configuration file `fs-server.conf` under “Web Applications” (see Chapter 4.3.1.7 page 46) and “WEBedit configuration” (see Chapter 4.3.1.14 page 61). Before using ContentCreator for productive operations, the configuration should be adapted to the respective requirements.

ContentCreator is also available as project local, configurable web application. This means that a separate ContentCreator instance can be installed, configured and activated for each project in the ServerManager (see Chapter 5.2.2 Page 191).

Certain prerequisites have to be fulfilled before using ContentCreator in FirstSpirit projects (see Chapter 5.2.1 page 190 and Chapter 5.2.2 page 191).

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 Chapter 4.6 page 138).

For information about functions, restrictions and expandability of the ContentCreator see also FirstSpirit Online Documentation, area “Plug-In Development” / “ContentCreator Extensions”.

5.2.1 Project prerequisites when using ContentCreator

The following basic requirements have to be fulfilled before using the ContentCreator mode in FirstSpirit projects. Check these requirements prior to application:

1. Partially, functions of input components in ContentCreator differ from that in SiteArchitect. For this reason, please check before use of ContentCreator if the input components required in the project are supported by ContentCreator. Also check customer-specific components properly.
2. Browser compatibility:
 - Which browser is used?
 - Which security settings are used in the company? Does ContentCreator run in this configuration?
3. Proxy / Firewall configuration: Is a proxy or firewall used? Does ContentCreator 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 ContentCreator users (if necessary also externally?)
5. Screen resolution: min.: 1024x768 pixel

5.2.2 ContentCreator as a local project application

ContentCreator is now also available as a configurable, local project web application. This means that a separate ContentCreator instance can be installed, configured and activated for each project via the ServerManager.

To do this, the "Web Components" area includes "ContentCreator" area. The tab can be used to configure a ContentCreator instance for the selected project (see Chapter 7.4.18 page 343).

Other web components can be added to the ContentCreator instance (e.g. FirstSpirit DynamicDatabaseAccess, FirstSpirit DynamicPersonalization). These web components can be individually configured for this web area ("ContentCreator"). Both default configuration using the "Configuration" button and manual editing of the "web.xml" file are possible.

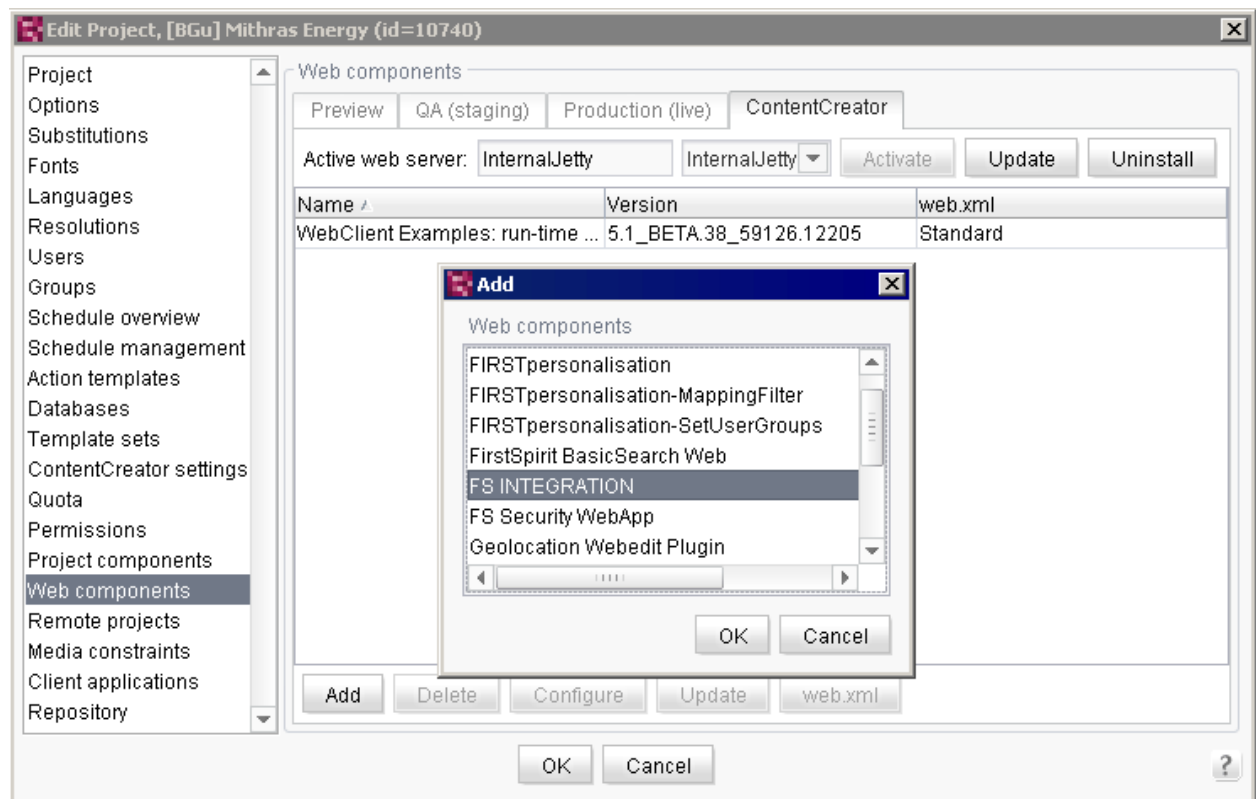


Figure 5-1: ContentCreator as a local project web application



The local project configuration enables an "advanced" ContentCreator 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 ContentCreator can now be combined with a project-specific personalization configuration for individual projects.

If the software on the server is updated, for example with a new ContentCreator version, it is possible to update all local project ContentCreator instances.

5.2.3 Browser configuration when using ContentCreator

Problems may arise while copying, cutting or pasting texts into the input component DOM editor (CMS_INPUT_DOM and CMS_INPUT_DOMTABLE) 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 ContentCreator 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");
```

When working with the web browser **Google Chrome** too, copying, cutting and pasting texts in DOM editors by using the respective context menu entries is disabled for security reasons. The



following message is displayed:

"Currently not supported by your browser, use keyboard shortcuts instead."

Use the following keyboard shortcuts instead:

- CTRL + C (Copy)
- CTRL + X (Cut)
- CTRL + V (Paste)

Please contact your system administrator if problems arise with the browser configuration.



6 FirstSpirit start page

Initial access to FirstSpirit Server is usually via the Internet. In accordance with the settings, the standard connection to FirstSpirit Server is established during installation (see the FirstSpirit installation instructions).

Depending on the login configuration, the login procedure may be automatic (see Chapter 6.1 page 194) or it may require entry of the user name and password (see Chapter 6.2 page 195). The login procedure is configured in the `fs-jaas.conf` configuration file (for information on the parameters, see Chapter 4.3.4 page 85).



Logging onto multiple FirstSpirit servers simultaneously with the same host names (e.g. myServer:8200 and myServer:8400) via a web browser is not supported.

6.1 Automatic login using single sign-on (SSO)

If the server has an SSO-compatible login module, the user can be authenticated (e.g. using his Windows login credentials) on the FirstSpirit server automatically. To do this, the `jaas.default` parameter must be configured for SSO (for configuration, see JAAS²³, Chapter 4.3.1.6 page 45). The corresponding configuration can be made using FirstSpirit ServerManager (see Chapter 7.3.10 page 264).

When the start page is retrieved, the system checks if automatic login is possible. If the server has an SSO-compatible login module, the user is logged onto the FirstSpirit server automatically using his Windows login credentials and is sent straight to the start page (see Chapter 6.3 page 196). If the user is not yet registered on the server under his Windows login credentials, he is added as a new external user.

²³ Java Authentication and Authorization Service

(for more information, see: <http://www.oracle.com/technetwork/java/javase/jaas/index.html>)



6.2 Login with user name and password

If automatic login fails (or if SSO login is not configured), a login page appears. The user can log onto the FirstSpirit server using the login screen. This login method applies to all applications on the server and will even retain inactive users for a certain length of time.

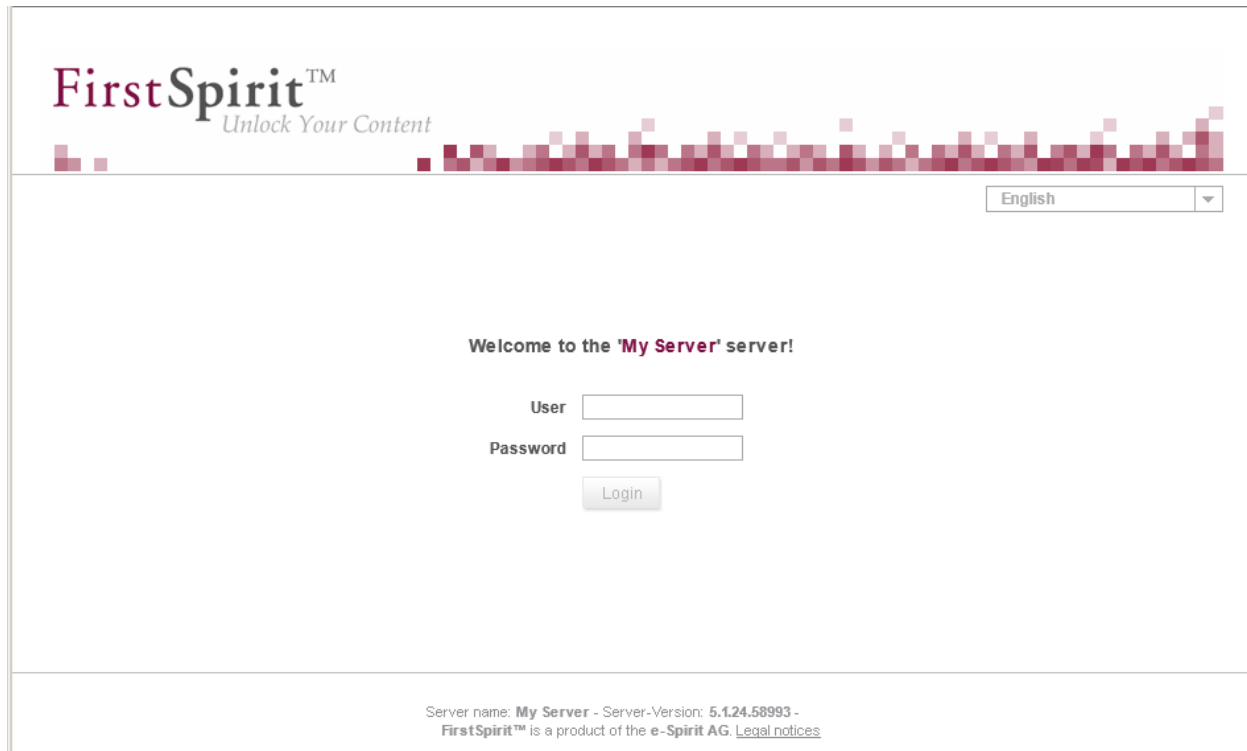


Figure 6-1: Login

This page contains information on the name and version of the FirstSpirit server:

Server name: name of the FirstSpirit server. If a symbolic name has been defined for the server in the `fs-server.conf` configuration file, this name appears on the start page (see Chapter 4.3.1.1 page 33). If no symbolic name has been defined, then the host name from the access path is shown; calling the server via <http://www.myServer.de:4050> therefore results in the server name "myServer", for instance.



This drop-down menu can be used to specify the language used when working with FirstSpirit.

Server version: the version is provided automatically by the server.

User: the user name that the user will use to log onto the FirstSpirit server is entered in this field.



Password: the user password is entered in this field.

Clicking on the "Login" button logs the user in under the registered user name.

6.3 FirstSpirit start page

After the user logs in (automatically or manually), the FirstSpirit start page appears. The start page is split into different areas:

- Start applications (see Chapter 6.3.1)
- Quick start (see Chapter 6.3.2)
- User (see Chapter 6.3.3)

To start FirstSpirit ServerManager and the FirstSpirit SiteArchitect, the Oracle Java Runtime Environment (JRE) is required, which contains Java Web Start (JRE is usually installed automatically during installation of JDK).

For more detailed information about the required version of JRE, please read the "Technical Data Sheet".



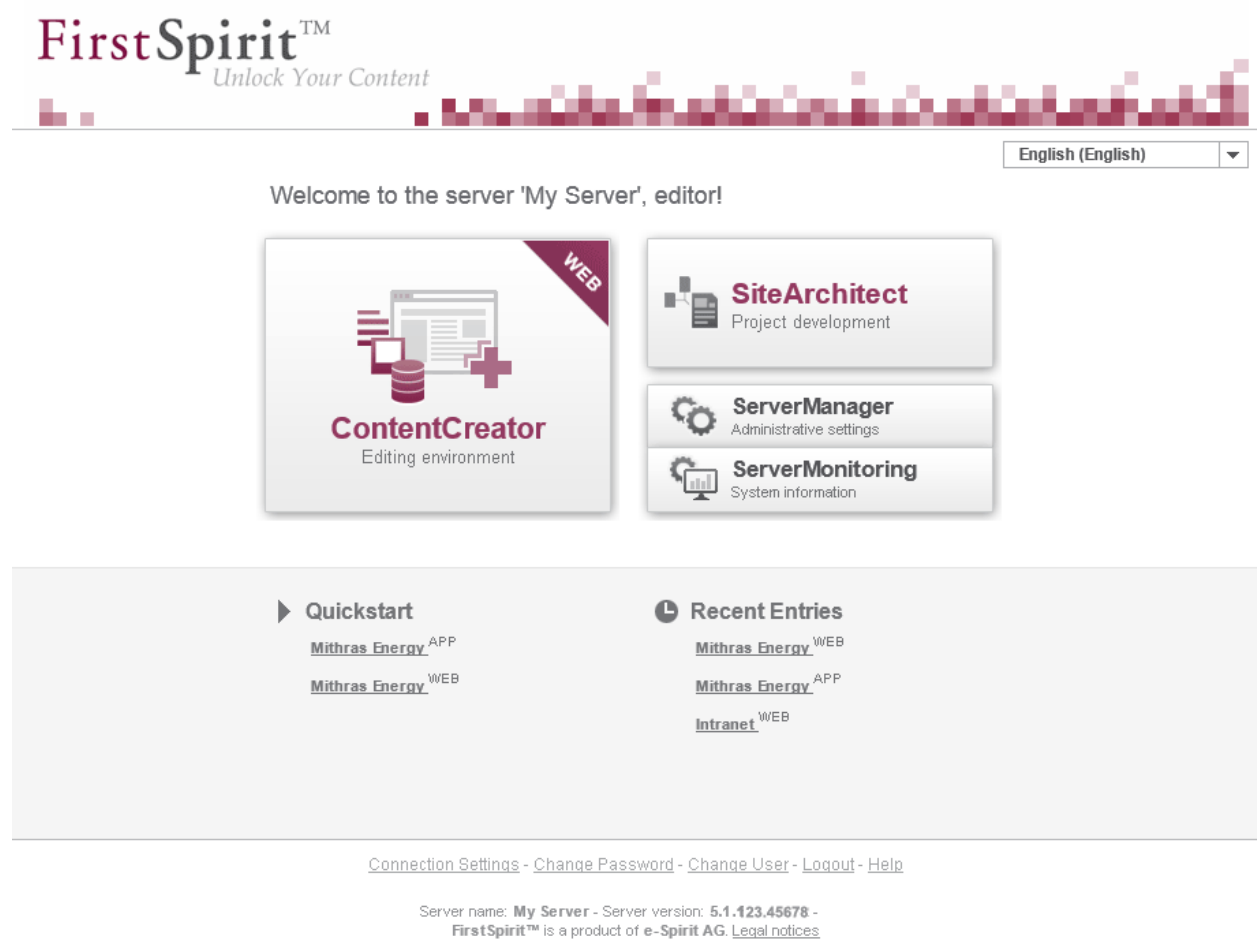


Figure 6-2: FirstSpirit start page

In addition, the page shows which user is currently logged in on the server.

 This drop-down menu can be used to specify the language used when working with FirstSpirit.

(To configure the FirstSpirit start page web application, see Chapter 5 page 189 ff.)

6.3.1 Starting applications

At the top of the page are entries for starting the FirstSpirit applications.

- **ContentCreator (Editing environment):** Click this entry to start the FirstSpirit ContentCreator editing system via a browser. The ContentCreator's range of functions



has been designed to cover editorial work in FirstSpirit projects (see Chapter 6.4.1).

- **SiteArchitect (Project development):** Click this entry to start FirstSpirit SiteArchitect. A project selection dialog appears, from which the user can select the desired project for editing. This project is then opened in SiteArchitect. A connection to the server will be established automatically (see Chapter 6.4.2).
- **ServerManager (Administrative settings):** Click this entry to open the ServerManager, which supports the configuration of FirstSpirit server and projects. Support is provided for both general administrative tasks carried out by a server administrator, as well as project-related administrative settings made by a project administrator. A detailed description of this is given in Chapter 7 (page 213).
- **ServerMonitoring (System information):** Click this entry to open FirstSpirit ServerMonitoring, which helps monitoring the FirstSpirit server. A detailed description of this is given in Chapter 8 (page 449).



ServerManager and ServerMonitoring can only be started by server and project administrators. Depending on the individual project configuration, the use of the ContentCreator can be deactivated.

6.3.2 Quick start

In the middle of the page are the Quickstart entries that are directly linked to a FirstSpirit project. Each project name appears next to the associated application (WEB = ContentCreator, APP = SiteArchitect), via which the relevant project is opened. Click the required project to start the associated application and open the selected project.

The area is divided into Quickstart projects and those which are Recent Entries.

- Under **Quickstart** are projects which have been configured in the ServerManager as Quickstart projects and which the logged-in user has permission to open. (To configure quick-start entries, see Chapter 7.3.8 page 261).
- Under **Recent Entries** are all projects which the logged-in user has recently edited.



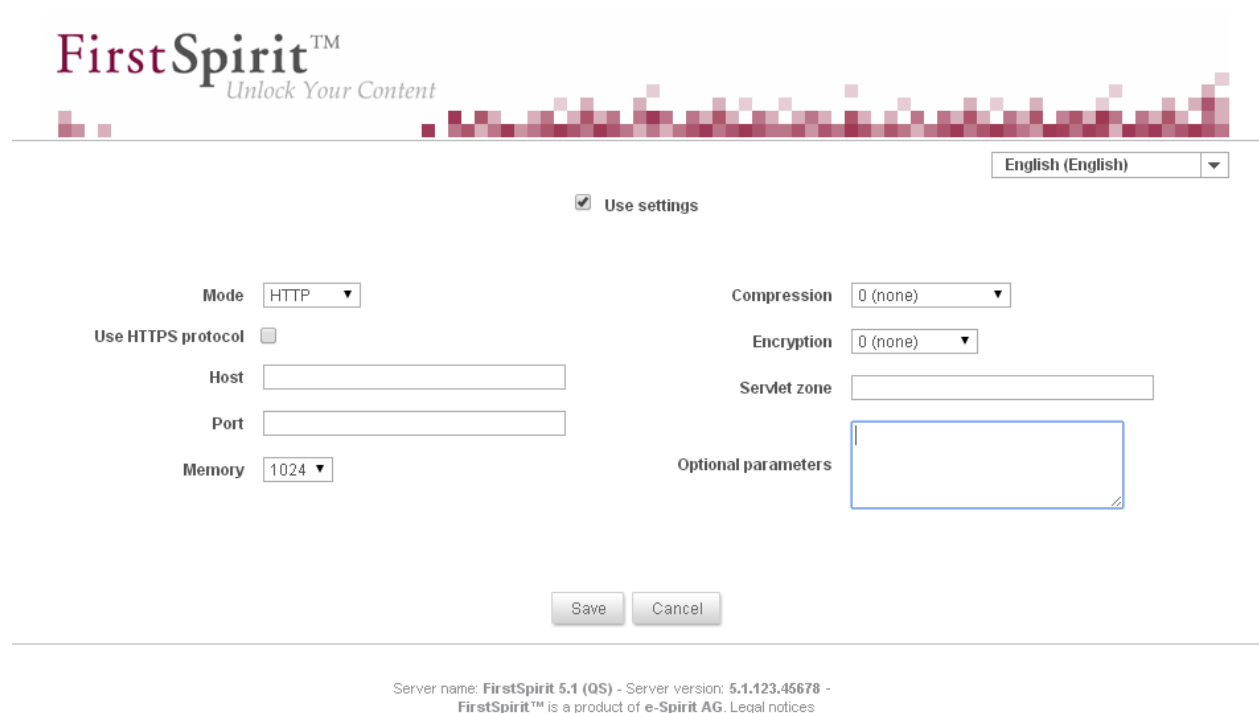
6.3.3 User

The bottom section includes entries for user settings related to the user who is currently logged in:

- **Connection settings:** here the connection settings for the user who is currently logged in can be changed (see Chapter 6.3.3.1 page 200).
- **Change password:** the password for the user who is currently logged in can be changed here (see Chapter 6.3.3.2 page 204).
- **Change user:** in some cases it may be desirable for a user to authenticate himself under a different user name on the FirstSpirit server in order to log on as a server administrator, for instance (see Chapter 6.3.3.3 page 205).
- **Logout:** clicking on this entry terminates the current FirstSpirit session for the user who is logged in (see Chapter 6.3.3.4 page 205).
- **Help:** clicking on this entry opens the FirstSpirit Online Documentation.



6.3.3.1 Configuring connection settings



The screenshot shows the 'FirstSpirit™ Unlock Your Content' header. Below it is a language selector set to 'English (English)'. A checkbox labeled 'Use settings' is checked. The configuration area includes:

- Mode:** A dropdown menu currently set to 'HTTP'.
- Use HTTPS protocol:** An unchecked checkbox.
- Host:** An empty text input field.
- Port:** An empty text input field.
- Memory:** A dropdown menu currently set to '1024'.
- Compression:** A dropdown menu currently set to '0 (none)'.
- Encryption:** A dropdown menu currently set to '0 (none)'.
- Servlet zone:** An empty text input field.
- Optional parameters:** A large empty text area.

At the bottom of the configuration area are 'Save' and 'Cancel' buttons. Below the buttons, the footer text reads: 'Server name: FirstSpirit 5.1 (QS) - Server version: 5.1.123.45678 - FirstSpirit™ is a product of e-Spirit AG. [Legal notices](#)'.

Figure 6-3: Configuring connection settings

Here the connection settings for starting SiteArchitect and using ServerManager are shown for the user who is currently logged in. The values configured here overwrite the server-wide Web Start settings for this user (see Chapter 7.3.7 page 260). The settings should only be changed for testing purposes.

Mode: this drop-down menu is used to set the connection mode for the standard communication between FirstSpirit clients and servers set for the user who is currently logged in:

- HTTP: normal Internet connection (default setting).
- Socket: direct connection mode.





To securely operate the FirstSpirit Server, we recommend that you run the complete client/server communication exclusively via HTTPS. This applies to both ServerManager and SiteArchitect.

While direct communication from these applications to the FirstSpirit Server is generally supported by the alternative socket mode, it is not deemed to be as secure as HTTPS communication.

Should you have questions about securely operating FirstSpirit, please do not hesitate to contact our Helpdesk.

Use HTTPS protocol: the parameter used to define whether communication will take place in HTTP connection mode using the secure HTTP protocol.

Host: server name or IP address of the FirstSpirit server to which the client is to connect during Web Start.

Port: FirstSpirit Server port number.

Memory: specifies the amount of memory (in MB) that will be made available for the client's virtual machine.

Compression: compression for the communication between FirstSpirit applications and servers set for the user who is currently logged in:

- None: no compression when transmitting data between clients and servers.
- Deflate: uses the deflate algorithm with standard compression for transmitting data between client and server.
- Deflate speed: uses the deflate algorithm with the fastest compression for transmitting data between client and server.
- Deflate best: uses the deflate algorithm with the best compression for transmitting data between client and server.
- Snappy: default setting for freshly installed FirstSpirit servers. "Snappy" compression mode is available to the Microsoft Windows (32 and 64 bit), Linux and Mac OS operating systems. The "Deflate speed" compression mode is used as the fallback mode (e.g. on operating systems that are not supported).



Encryption: encryption for the communication between FirstSpirit applications and servers set for the user who is currently logged in:

- None: no encryption when transmitting data between clients and servers.
- TLS²⁴: uses the TLS protocol for transmitting data between client and server.
- DH ARC4: uses the DH ARC4 encryption algorithm for transmitting data between client and server.

Servlet zone: information on the servlet zone (the servlet zone corresponds to mapping of the root application URL (for `WEBAPP_ROOT_URL` parameter information, see Chapter 4.3.1.7 page 46)).

Optional parameters: optional parameters for Web Start configuration can be stored in this field. The parameters can be specified sequentially in the following format, separated by a semicolon: `PARAMETER1=VALUE1; PARAMETER2=VALUE2`. The optional parameters may correspond to the settings options that can be entered in the dialog's input fields (e.g., refer to Encryption, Compression):

- `compression`: parameter for compression (for communication between FirstSpirit applications and servers).
Possible values: 0 (none), 1 (Deflate), 2 (Deflate speed), 3 (Deflate best)
Example: `compression=3`
- `encryption`: parameter for encryption (for communication between FirstSpirit applications and servers).
Possible values: 0 (none), 1 (TLS), 2 (DH ARC4)
Example: `encryption=1`
- `login`: login module information (plain, sso). Example: `login=plain`
- `autologin`: parameter for user login and password information. Both values are passed as plain text, separated by ":". The password can be left out, but the colon must always be passed.
- `host`: host name information (server pre-selection).
- `port`: port number information (integer).
- `mode`: connection mode information (`http`, `socket`)
- `httpproxy`: parameter for information on the proxy to be used in HTTP connection mode. If this parameter is specified, only this proxy will be used. If the parameter is not specified, Java

²⁴ Transport Layer Security



Web Start will attempt to analyze the proxy configuration. This analysis can be prevented by entering the `nohttpproxy=1` parameter.

- `httpsproxy`: parameter for information on the proxy to be used in socket connection mode. The proxy in this case is used for tunneling TCP connections.
- `nohttpproxy`: parameter used to prevent analysis of the Java Web Start proxy configuration (see section `httpproxy`).

Example: `nohttpproxy=1`

- `usehttps`: parameter used to define whether the communication is to be in HTTP connection mode using the secure HTTPS protocol (value=1) or not (value=0).

Example: `usehttps=1`

- `proxybypass`: parameter used to define which hosts may bypass the proxy. In HTTP mode, communication (for these hosts) is then not via a proxy. Multiple hosts can be passed as a semicolon-separated list. When entering one (or more) host names, all hosts that start with the specified host name (see, for example, `myServer_1`, `myServer2`, etc.) may bypass the proxy.

Example: `proxybypass=myServer;localhost`

- `inlinebrowser.httpproxy`: parameter used to configure the inline browser (for integrated preview) for communication via an HTTP proxy (similar to the proxy configuration of the FirstSpirit SiteArchitect). The server name or IP address of the proxy and the port must be specified for this. An HTTP proxy configuration set using this parameter temporarily overrides the browser's local configuration settings, but it is not stored there.

Example: `inlinebrowser.httpproxy=myServer:8888`

- `CLIENT_HOME_DIR`: parameter used for information on the directory in the file system in which the client applications are to be stored (see Chapter 4.3.1.2 page 37, and Chapter 4.10.2 page 185).
- `disableExpensiveCSOperations`: this parameter is used to disable the following operations for data sources that may require a lot of memory and time if they have a large number of data records (e.g. 1 million data records):
 - "Simple search" in the search dialog box of a data source
 - "Advanced search" in the search dialog box of a data source
 - "Full-text search" in the search dialog box of a data source
 - "Display all data records" operation in the data source view and selection dialog box of the input components (e.g. `FS_DATASET`)
 - Sorting function covering the columns in the data source overview and selection dialog covering the input components (e.g. `FS_DATASET`)
- To deactivate these operations for **all** data sources, the parameter must be entered without a value:

```
disableExpensiveCSOperations=
```



- If the operations are to be deactivated only for **individual** data sources, the unique identifiers of the data sources must be provided, separated by commas:

```
disableExpensiveCSOperations=datenquelle1,datenquelle2
```

- Deactivating individual data sources does not affect the use of the (legacy) input components CMS_INPUT_CONTENTLIST, CMS_INPUT_OBJECTCHOOSER and CMS_INPUT_TABLIST. If limitations apply to these input components as well, they must be expanded to include all data sources.

Clicking the **Save** button saves the changed connection settings for the user who is currently logged in. To activate the settings, the checkbox "Apply settings" must be selected. The information "Connection Settings (enabled)" then appears on the start page.

6.3.3.2 Change password

Clicking on this entry allows the user to change his FirstSpirit Server login password.

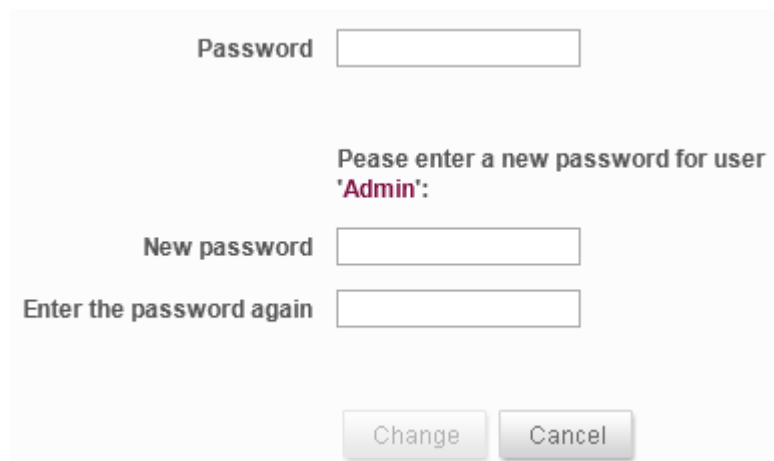


Figure 6-4 shows a "Change password" dialog box. It features three input fields: "Password", "New password", and "Enter the password again". Above the "New password" field, there is a prompt: "Pease enter a new password for user 'Admin':". At the bottom of the dialog, there are two buttons: "Change" and "Cancel".

Figure 6-4: Change password

Password: the current password must be entered again in this field.

New password: the new password is entered in this field.

Repeat password: the new password is entered again in this field to rule out any accidental spelling errors when changing the password.

Clicking on the "Change" button applies the new password to the user who is logged in.





This entry is available only to users who have logged onto the server manually and not to external users who have been added using an automatic SSO login.

6.3.3.3 Change user

Clicking on this entry allows a different user to log on to the server. The login page appears again (see Chapter 6.2, starting on page 195) and now also includes the automatic login option.

FirstSpirit™
Unlock Your Content

English (English)

You have logged out successfully. Please log in again.

Please use the "Automatic Login" button to log in automatically without entering your user name and password.

Automatic Login

User

Password

Login

Server name: FirstSpirit 5.1 (QS) - Server version: 5.1.123.45678
FirstSpirit™ is a product of e-Spirit AG. [Legal notices](#)

Figure 6-5: Java Web Start – Changing users

If the server has an SSO-compatible login module, there will now be an option to log onto the server automatically using Windows login credentials. Clicking on the "Automatic Login" button logs the user in using Windows login credentials.

6.3.3.4 Logout

Clicking on this entry lets the user log out of the system. The login page shown in Figure 6-5 then reappears (see also Chapter 6.2, starting on page 195).



6.4 Starting the applications

6.4.1 ContentCreator

ContentCreator does not require a Java environment and can be accessed directly via the web browser. To start the client, simply click on the relevant entry on the FirstSpirit start page (see Chapter 6.3 page 196).

When starting ContentCreator, a project selection dialog appears after a connection is established; the dialog contains a list of projects available to the user who is logged in. To appear in this list a project must be activated for editing in ContentCreator (see Chapter 7.4.14 page 330). Only the sample project is available initially, unless it was deselected during the FirstSpirit Server installation. Initially, only the sample project is available after installation, unless it was deselected during the FirstSpirit Server installation.

For detailed information about FirstSpirit ContentCreator please see also respective documentation.

6.4.2 SiteArchitect

To start SiteArchitect, a web browser that has "Java Web Start"²⁵ is required. Java Web Start is used to pass FirstSpirit product software updates to client systems automatically when started²⁶. Among other things, the required permissions (e.g. file creation rights) must be configured at the system or user level.

To start the application, simply click on the relevant entry on the FirstSpirit start page (see Chapter 6.3 page 196).

Note: Additional messages and information may be displayed depending on the browser used. For example, when starting SiteArchitect for the first time, Google Chrome requires confirmation from the user to run Java Webstart for this file type. To prevent this message from being displayed during future starts, you can open the context menu for the corresponding message and select the entry "Always open files of this type" (if that is desired).

²⁵ For more information, see: <http://www.oracle.com/technetwork/java/javase/javawebstart/index.html>

²⁶ Principle: http://de.wikipedia.org/wiki/Java_Web_Start



When starting SiteArchitect, a project selection dialog appears after a connection is established; the dialog contains a list of projects available to the user who is logged in. Initially, only the sample project is available after installation, unless it was deselected during the FirstSpirit Server installation.

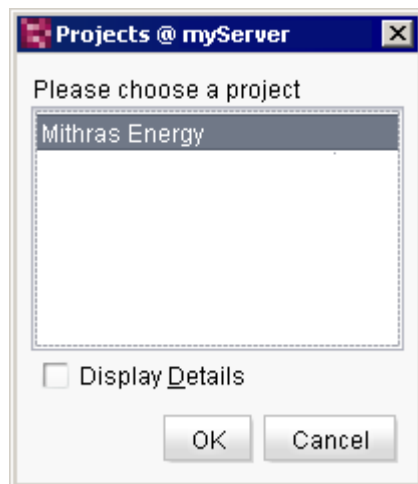


Figure 6-6: Selecting a project

Double-click on the entry or click on the "OK" button to load the selected project.

Now you can start familiarizing yourself with FirstSpirit Client.

For detailed information about FirstSpirit SiteArchitect please see also respective documentation.



Launch problems with Mac OS: The aim of the "Gatekeeper" security function on Apple Macintosh systems were reset to the "Mac App Store" default value with version 10.9.5. When the FirstSpirit SiteArchitect or ServerManager is launched, the following message may be displayed: "FIRSTspirit.jnlp can't be opened because it is from an unidentified developer."

To enable SiteArchitect and ServerManager to be restarted, the settings must be reset to "Anywhere" under

"Apple menu / System Preferences / Security & Privacy / General / Allow applications downloaded from".

All FirstSpirit versions 4.x and 5.x are affected by this problem.



6.4.3 ServerManager

To start ServerManager, a web browser that has "Java Web Start"²⁷ is required. Java Web Start is used to pass FirstSpirit product software updates to client systems automatically when started²⁸. Among other things, the required permissions (e.g. file creation rights) must be configured at the system or user level.

To start the application, simply click on the relevant on the FirstSpirit start page (see Chapter 6.3 page 196).

Note: Additional messages and information may be displayed depending on the browser used. For example, when starting SiteArchitect for the first time, Google Chrome requires confirmation from the user to run Java Webstart for this file type. To prevent this message from being displayed during future starts, you can open the context menu for the corresponding message and select the entry "Always open files of this type" (if that is desired).

6.4.4 ServerMonitoring

ServerMonitoring does not require a Java environment and can be accessed directly via the web browser. To start the application, simply click on the relevant entry on the FirstSpirit start page (see Chapter 6.3 page 196). This is where the license file can be easily installed, for instance (see Chapter 8.6.1.2 page 471).

6.5 Starting FirstSpirit Client as a Java application

If the FirstSpirit server is not accessed using a command line instead of over the Internet, parameters for communication between SiteArchitect and the FirstSpirit server can be configured in the following connection dialog, similarly to how the connection settings were configured in Chapter 6.3.3.1, starting on page 200.

²⁷ For more information, see: <http://www.oracle.com/technetwork/java/javase/javawebstart/index.html>

²⁸ Principle: http://de.wikipedia.org/wiki/Java_Web_Start



6.5.1 Socket mode

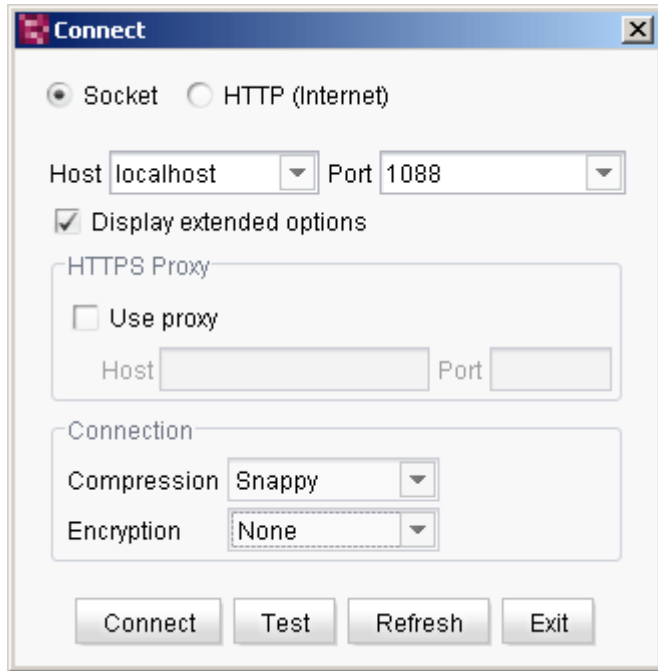


Figure 6-7: SiteArchitect connection dialog – Socket mode

Host: server name or IP address of the FirstSpirit server to which the client is to connect during Web Start.

Port: FirstSpirit server port number.

If the checkbox "Display extended options" is selected, the following parameters can be set:

Use proxy: if this checkbox is selected, a proxy server to be used for communication between SiteArchitect and the FirstSpirit server can be defined in the "Host" and "Port" fields below it.

Compression: compression for the communication between FirstSpirit applications and servers set for the user who is currently logged in:

- **None:** no compression when transmitting data between clients and servers.
- **Deflate:** uses the deflate algorithm with standard compression for transmitting data between client and server.
- **Deflate_Speed:** uses the deflate algorithm with the fastest compression for transmitting data between client and server.



- **Deflate_Best:** uses the deflate algorithm with the best compression for transmitting data between client and server.
- **Snappy:** default setting for freshly installed FirstSpirit servers. "Snappy" compression mode is available to the Microsoft Windows (32 and 64 bit), Linux and Mac OS operating systems. The "Deflate speed" compression mode is used as the fallback mode (e.g. on operating systems that are not supported).

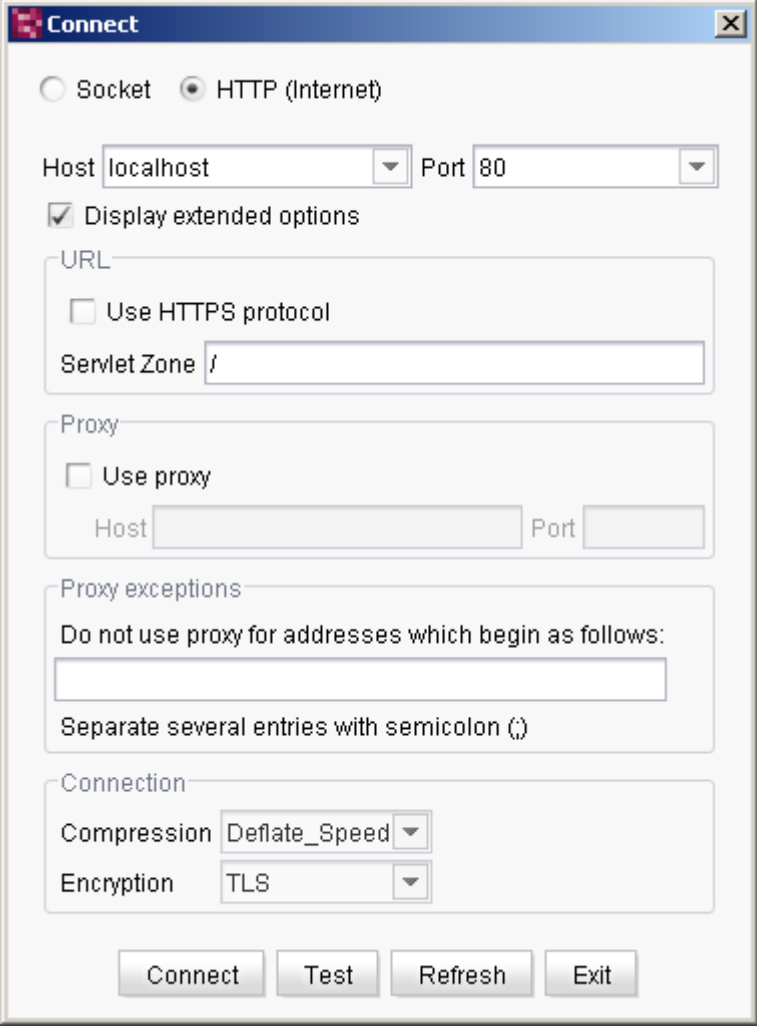
Encryption: encryption for the communication between FirstSpirit applications and servers set for the user who is currently logged in:

- **None:** no encryption when transmitting data between clients and servers.
- **TLS:** uses the TLS protocol for transmitting data between client and server.
- **DH_Arc4:** uses the DH ARC4 encryption algorithm for transmitting data between client and server.

For a description of the settings, see Chapter 6.3.3.1 page 200.



6.5.2 HTTP (Internet) mode



The screenshot shows the 'Connect' dialog box with the 'HTTP (Internet)' radio button selected. The 'Host' field is set to 'localhost' and the 'Port' field is set to '80'. The 'Display extended options' checkbox is checked. The 'URL' section contains a 'Servlet Zone' field with a '/' character. The 'Proxy' section has the 'Use proxy' checkbox unchecked. The 'Proxy exceptions' section has a text box for addresses to exclude. The 'Connection' section has 'Compression' set to 'Deflate_Speed' and 'Encryption' set to 'TLS'. At the bottom are buttons for 'Connect', 'Test', 'Refresh', and 'Exit'.

Figure 6-8: SiteArchitect connection dialog – HTTP mode

For HTTP mode, a host name and port number different from the one for socket mode can be specified. The following parameters can also be configured:

Use HTTPS protocol: if this checkbox is selected, communication between client and server is encrypted using the HTTPS protocol.

Servlet zone: path to servlet directory. The path must always start with a "/".

Do not use proxy for addresses which begin as follows: information on domains that are to be called directly and not over a proxy connection, e.g. addresses for dedicated company networks. A semicolon is used to separate multiple addresses.



In both cases (socket and HTTP), the SiteArchitect/FirstSpirit server connection can be tested with the settings made using the "Test" button. If a connection is not possible, the configuration will need to be changed. If the test was successful, the connection can be established by clicking the "Connect" button.

For a description of the settings, see Chapter 6.3.3.1 page 200.



7 FirstSpirit ServerManager

The FirstSpirit ServerManager is a Java application with a convenient, swing-based user interface which supports the FirstSpirit administrator for general, administrative FirstSpirit tasks. The user interface can be used to create and configure new FirstSpirit projects. Besides the general tasks, it provides extensive functions. The ServerManager can, e.g., be used to define users or integrate existing identity management systems, such as LDAP or Active Directory. Analogue to the SiteArchitect, the ServerManager is started and updated via Java Web Start.

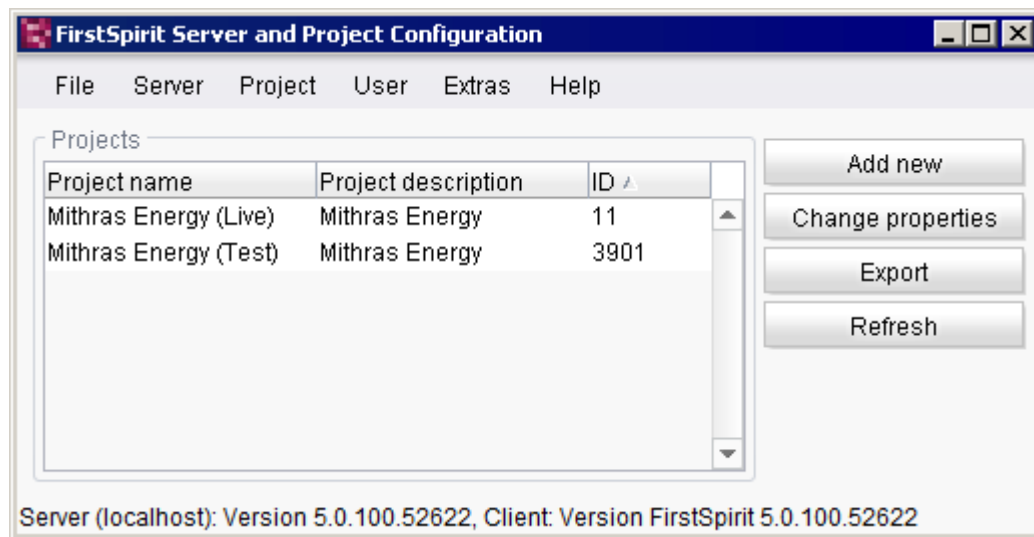


Figure 7-1: ServerManager

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.

The user with the ID 1 who was automatically created with ID 1 when a FirstSpirit Server was installed is named **Administrator**. There is only one administrator per Server. In principle, he has got all rights for FirstSpirit Server and applications. However, this user can be blocked from accessing specific projects ("ServerManager" / "Project" / "Settings" / "Options" / "Block administrator"). He is always server administrator at the same time, this role cannot be taken away from this user (he is "super administrator").



Server administrators always hold all of the permissions of the administrator. The role of server administrator can only be assigned to a FirstSpirit user by the administrator or by another server administrator. There can be multiple server administrators per server.

Administrators and server administrators can for example:

- Create / export / import / 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

Project administrators always hold all permissions in projects in which they have been added to the default "Administrators" group (see Chapter 7.4.8.6 page 321).

By default, they can

- Change the properties of their project
- Export their project
- Clean up their project

Server and project administrators have got all permissions by default in newly created projects in **SiteArchitect**.

For more information please see Chapter 7.2.4 page 235.

7.2 Menu bar items

The following describes the individual items in the menu bar and ServerManager.

7.2.1 File

7.2.1.1 Exit

This function closes ServerManager



7.2.2 Server

7.2.2.1 Clean-up

This function is used to purge files such as the backup and log files. The following cleanup activities can be carried out using the "Clean up server" dialog:

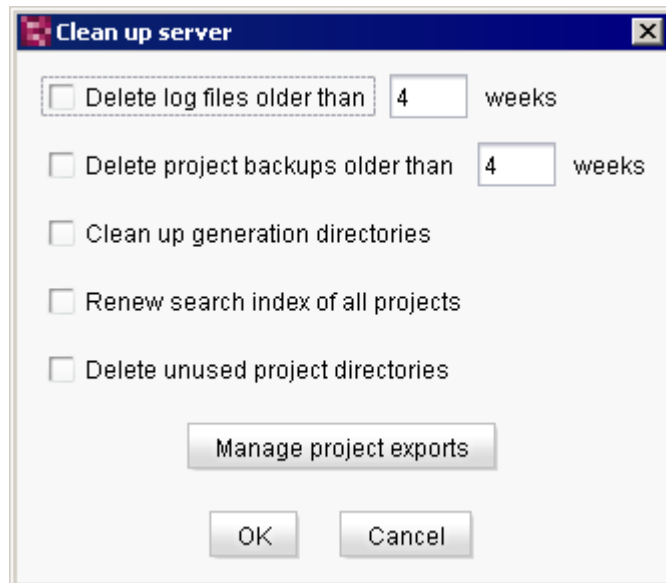


Figure 7-2: Clean up server

Delete log files: Deletes all files in the server log directory that are older than the specified number of weeks.

Delete project backups: Deletes all (automatically created) backups of projects that are older than the specified number of weeks. All compressed export files in the "backup" subdirectory directly under the FirstSpirit server root directory are affected. These files are created inside Schedule entry planning (see section 7.5.10.3, page 412). (In this case it is possible to move the directory to a different hard disk for backup (`BACKUP_PATH` parameter, see section 4.3.1.15, page 62).)



Backups created automatically follow a particular naming convention, such as '20130417_103642_projectname.tar.gz' – This action will not remove backups that do not follow this convention.





In the export directory for projects, enabling the function "Delete project backups" does not delete any projects.

Enterprise backups: In addition to the project backups created automatically, the action also removes all enterprise backup files. An "enterprise backup" is a license-dependent function. For more information, refer to the "Enterprise Backup" FirstSpirit module documentation.

The following rule applies to enterprise backups: at a minimum, the last snapshot is always required for the backup to function completely. Additional differential and incremental backups can be created based on a snapshot. If all files (snapshots, differential and incremental backups) are present within the cleanup period (older than x weeks), all files will be deleted completely. If, however, at least the last incremental or differential backup is present within the period and is not to be deleted, all differential and incremental backups, including the related snapshots, will not be deleted, making it possible to restore this backup set. However, all snapshots and differential and incremental backups that were present before the last snapshot will be deleted.

Clean up generation directories: This function deletes obsolete files from deleted and deactivated projects. The following directories and their content that were created during generation and deployment are affected.

- Cleans up the "project_projectID" directories in the FirstSpirit server "web/fs5staging" subdirectory.
- Cleans up the "project_projectID_partial_deployment" directories in the FirstSpirit server "web/fs5staging" subdirectory.
- Cleans up the directories containing the project previews ("project_projectID" directory in the "preview_cache" of the "web/fs5staging" subdirectory). The preview directory and its paths can be configured using the fs_server.conf configuration file (see section 4.3.1.8, page 49).

Renew search index of all projects: FirstSpirit features the ability to perform a search (full text search using the Lucene search index) using SiteArchitect selection dialogs. To do this, the search index must be calculated for the project. If this option is selected, indexing is carried out for searching all projects on the server. In this case only the current state of the elements in the project are taken into account. Historic data (e.g. changes to an object within a certain period) are not included in the calculation. Rebuilding the search index may take a long time, depending on the quantity and size of the projects. Individual projects or subareas of a project can be indexed using the FirstSpirit Access API (see Chapter 7.5.10.5 page 412).



For more information, see section 9.18, page 516.

Delete unused project directories: In some cases projects cannot be deleted completely, for instance when the operating system is still accessing files in the project directories. These projects can no longer be accessed using the ServerManager application. The associated project directories can be deleted completely using this option.

Manage project exports: Clicking this button opens the "Manage project exports" dialog:

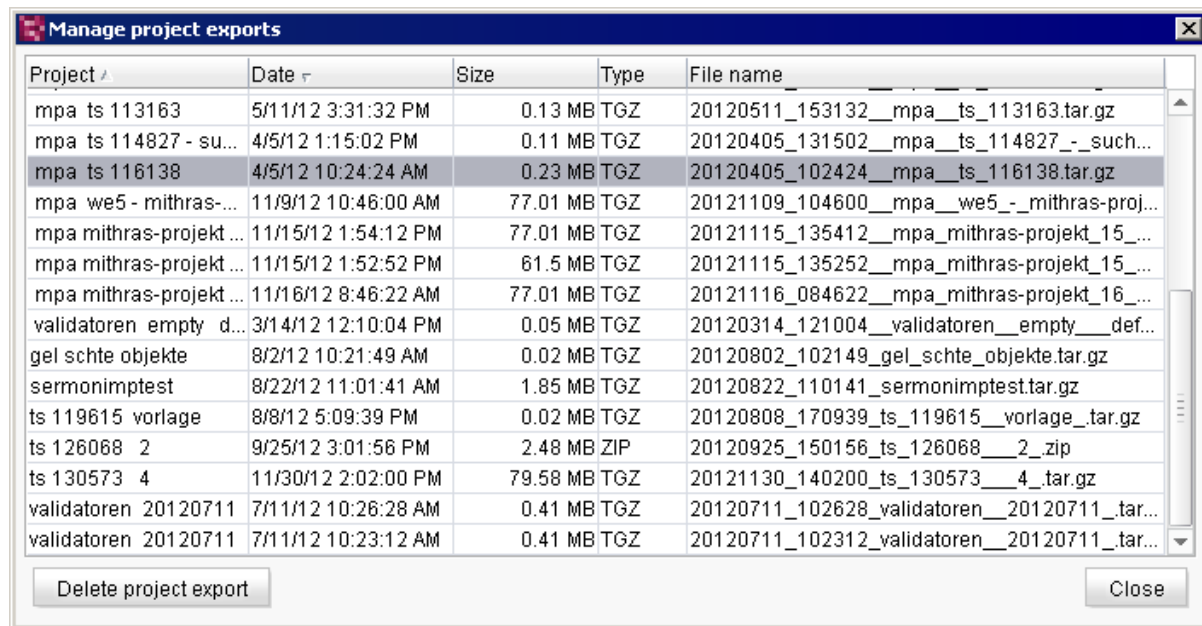


Figure 7-3: Manage project exports

This dialog lists all existing projects in table form. These projects can be marked in the overview and deleted if necessary by clicking "Delete project export".

7.2.2.2 Shutdown

After confirming the security prompt asking whether you really want to shut down the server, the ServerManager dialog is closed using this function and the server is shut down.



Restarting the server using ServerManager is NOT possible.

7.2.2.3 Properties

Calling this function opens a window in which a whole series of server properties can be modified.

A more detailed description of these server properties can be found in Chapter 7.3, starting on page 243.

7.2.3 Project

7.2.3.1 Add new

Use this function to create a new project on the server.

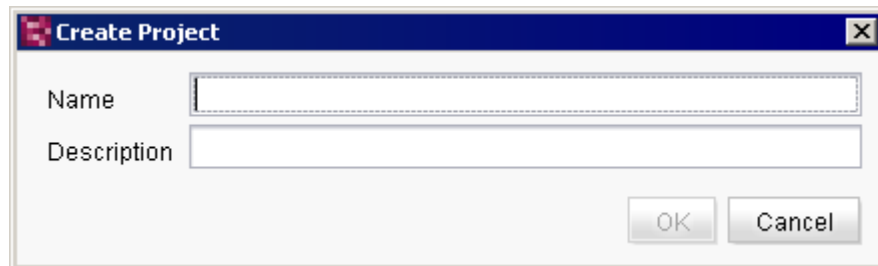


Figure 7-4: Project – Create

Name: The assigned project name must be unique server wide for being able to – especially in the case where there is more than one project on the Server – identify it afterwards. It is helpful to assign a meaningful name. Amongst others, the name will be displayed in the headline of the SiteArchitect.

Description: the description specified here appears in the project selection list after the user logs in.

After entering the project name and project description, the project will then be defined in the next dialog.



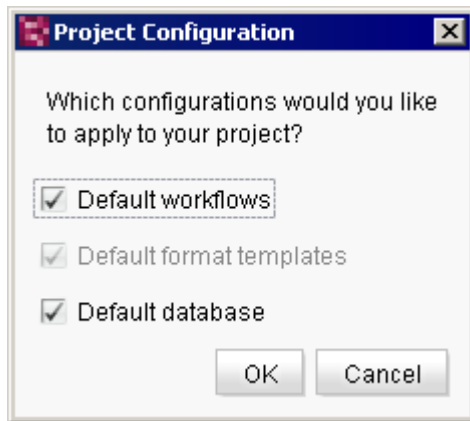


Figure 7-5: Project configuration

Selecting these checkboxes applies the respective configurations to the project.

Default format templates: the default format templates included with FirstSpirit are imported to the format template node (within the project's template store). Editors are also able to prepare text in bold or italics in the DOM Editor input component. This option cannot be deselected.

Default workflows: if this checkbox is selected, the default workflows included with FirstSpirit are imported to the workflow node (within the project's template store). There are two workflows integrated in FirstSpirit:

1. "Task": workflow for general editing of tasks in the project.
2. "Release request ": workflow for release of objects that were created or changed in the project.

Default database: if this checkbox is selected, the default database (Derby) provided with FirstSpirit is activated for the project. The standard layer can be used in FirstSpirit SiteArchitect for a database schema. This automatically sets write access to the database for this project (see Chapter 7.4.12 page 325).



The Derby DBMS contained in FirstSpirit is not suited for production mode and therefore should only be used for testing purposes.

Clicking "OK" to confirm the configuration will immediately add the new project to the ServerManager project list; the project can then be edited in more detail using the "Change properties" button (see Chapter 7.3.13, starting on page 278).



The "Add new" button offers the same functionality.

This menu item is only available to server administrators.

7.2.3.2 Import

Using the Import Project dialog, you can reimport projects to the server that had been exported.

This menu item is only available to server administrators.

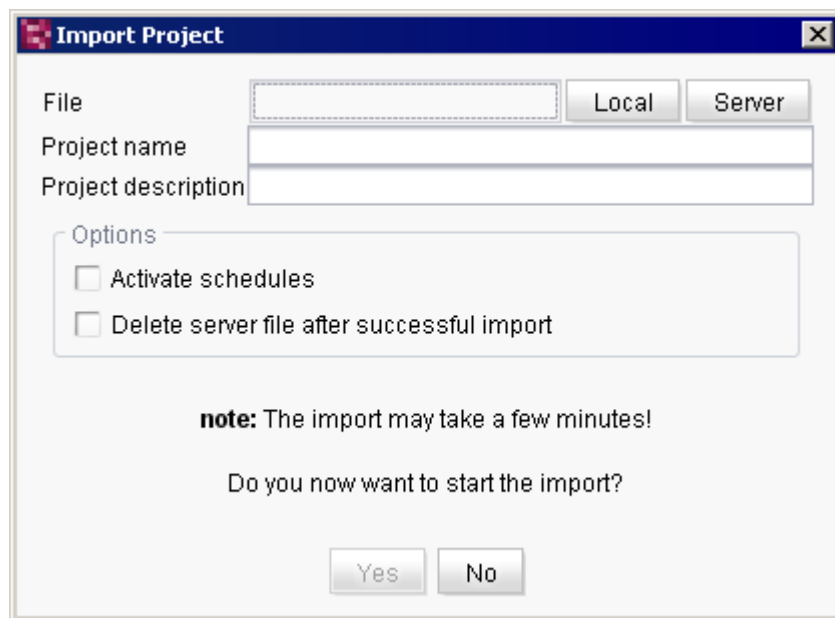
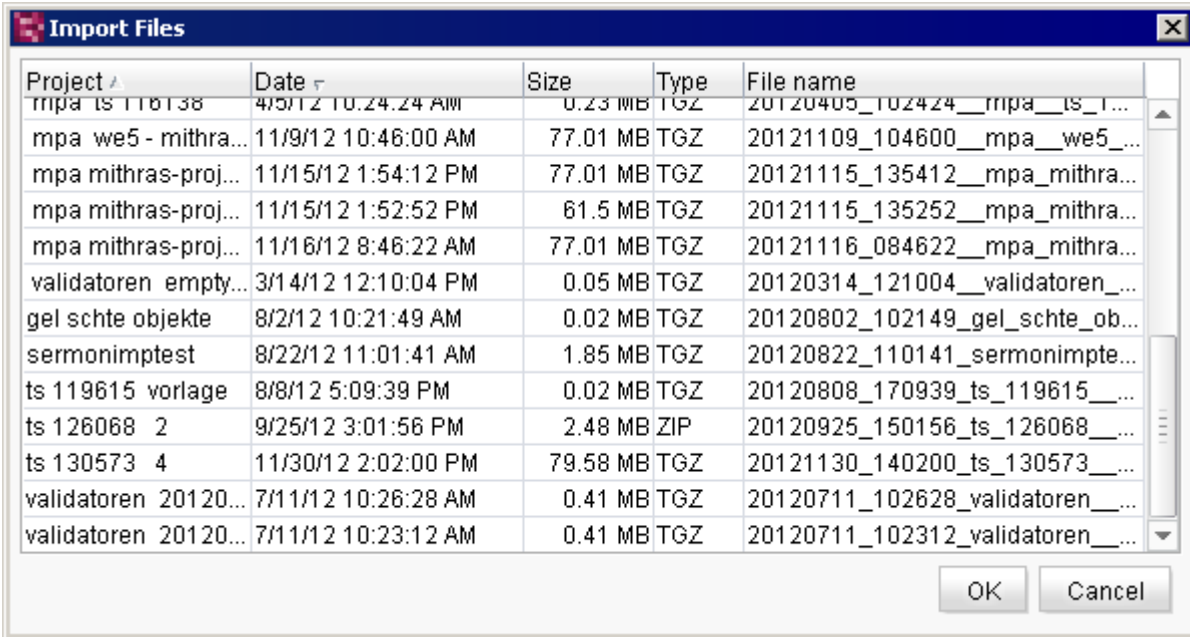


Figure 7-6: Project – Import

File: the Local and Server buttons can be used to select a compressed project export file. The **Local** button is used to search through the computer file system.

The **Server** button is used to view a list of packed files from the server's export directory. The desired project can be selected from this list (see Figure 7-7):





Project	Date	Size	Type	File name
mpa ts 116138	4/5/12 10:24:24 AM	0.23 MB	TGZ	20120405_102424__mpa_ts_1...
mpa we5 - mithra...	11/9/12 10:46:00 AM	77.01 MB	TGZ	20121109_104600__mpa_we5_...
mpa mithras-proj...	11/15/12 1:54:12 PM	77.01 MB	TGZ	20121115_135412__mpa_mithra...
mpa mithras-proj...	11/15/12 1:52:52 PM	61.5 MB	TGZ	20121115_135252__mpa_mithra...
mpa mithras-proj...	11/16/12 8:46:22 AM	77.01 MB	TGZ	20121116_084622__mpa_mithra...
validatoren empty...	3/14/12 12:10:04 PM	0.05 MB	TGZ	20120314_121004__validatoren_...
gel schte objekte	8/2/12 10:21:49 AM	0.02 MB	TGZ	20120802_102149_gel_schte_ob...
sermonimptest	8/22/12 11:01:41 AM	1.85 MB	TGZ	20120822_110141_sermonimpte...
ts 119615 vorlage	8/8/12 5:09:39 PM	0.02 MB	TGZ	20120808_170939_ts_119615_...
ts 126068 2	9/25/12 3:01:56 PM	2.48 MB	ZIP	20120925_150156_ts_126068_...
ts 130573 4	11/30/12 2:02:00 PM	79.58 MB	TGZ	20121130_140200_ts_130573_...
validatoren 20120...	7/11/12 10:26:28 AM	0.41 MB	TGZ	20120711_102628_validatoren_...
validatoren 20120...	7/11/12 10:23:12 AM	0.41 MB	TGZ	20120711_102312_validatoren_...

Figure 7-7: Import project – Import Files list

Project name: a unique name for the project to be imported must be entered in this field. See also Chapter 7.2.3.1 page 218.

Project description: a project description for the project to be imported must be entered in this field. See also Chapter 7.2.3.1 page 218.

Options

Activate schedules

If this checkbox is **selected**, all schedule entries (see Chapter 7.4.10 page 323) remain in the same status that they were in at the time the project was exported. This means that active schedule entries continue to remain active and are executed at the configured time. This option should therefore only be activated with caution and when the entries are known, since otherwise undesirable deployments may result. If this checkbox is **not selected**, the schedule entries for a project are preserved, but are deactivated.

Delete server file after successful import

If this checkbox is **selected**, the project export file is deleted from the server after the project is imported. It will no longer be available in the File Import list (see Figure 7-7).

The import is started by clicking "Yes".



If the project to be imported was created in a later version of FirstSpirit Server, the following warning will appear after the file is analyzed:

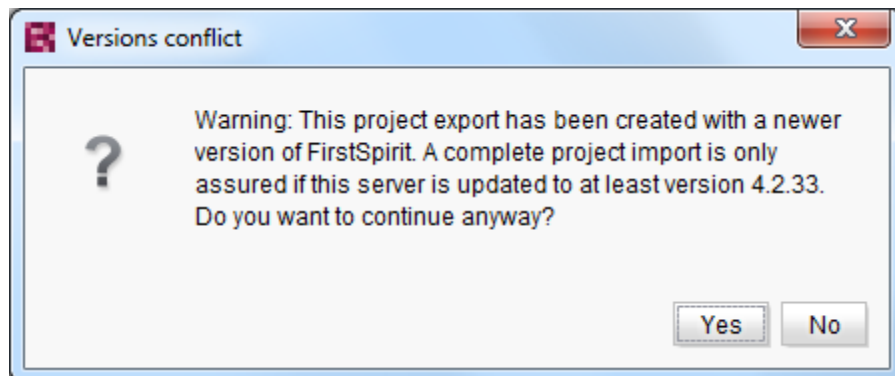


Figure 7-8: Version conflict during import

Clicking "No" will cancel the import process. Clicking "Yes" will continue the import.



There is no guarantee that importing projects that were created with a later version of FirstSpirit Server will work! This type of export/import is technically possible, but may result in unpredictable problems due to changes in software behavior.

If database content is used within the project, this content must be mapped to a new database layer before importing the project. When importing a project, a new or existing database layer on the server will have to be assigned to each schema of the project (see Figure 7-9):



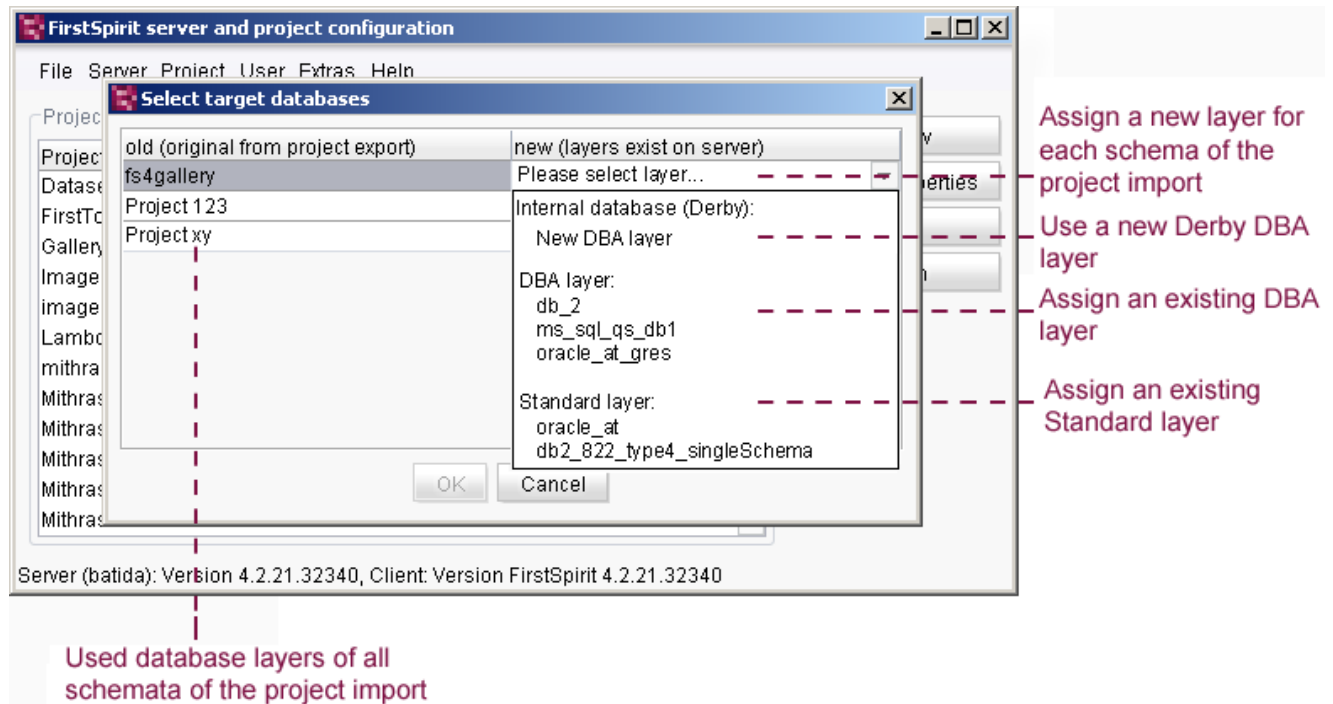


Figure 7-9: Select target databases

All schemata existing in the project export file are listed in the "Select target databases" dialog (left-hand side). The following options are listed on the right-hand side of the dialog to assign a new layer:

- New Derby DBA layer
- Standard layer
- DBA layer



The Derby DBMS contained in FirstSpirit is not suited for production mode and therefore should only be used for testing purposes.

The layer type selected determines whether FirstSpirit users can personally add new schemata to the project after the import (possible with DBA layers) or not (not permitted for standard layers) (For more information, see Chapter 4.9.4.2 page 171).



More detailed information on the different layer types is available in the FirstSpirit Manual for Developers (Basics).



The import process may take several minutes, depending on the size of the project. The progress of the entire import process, including individual steps, is displayed in the Import dialog.

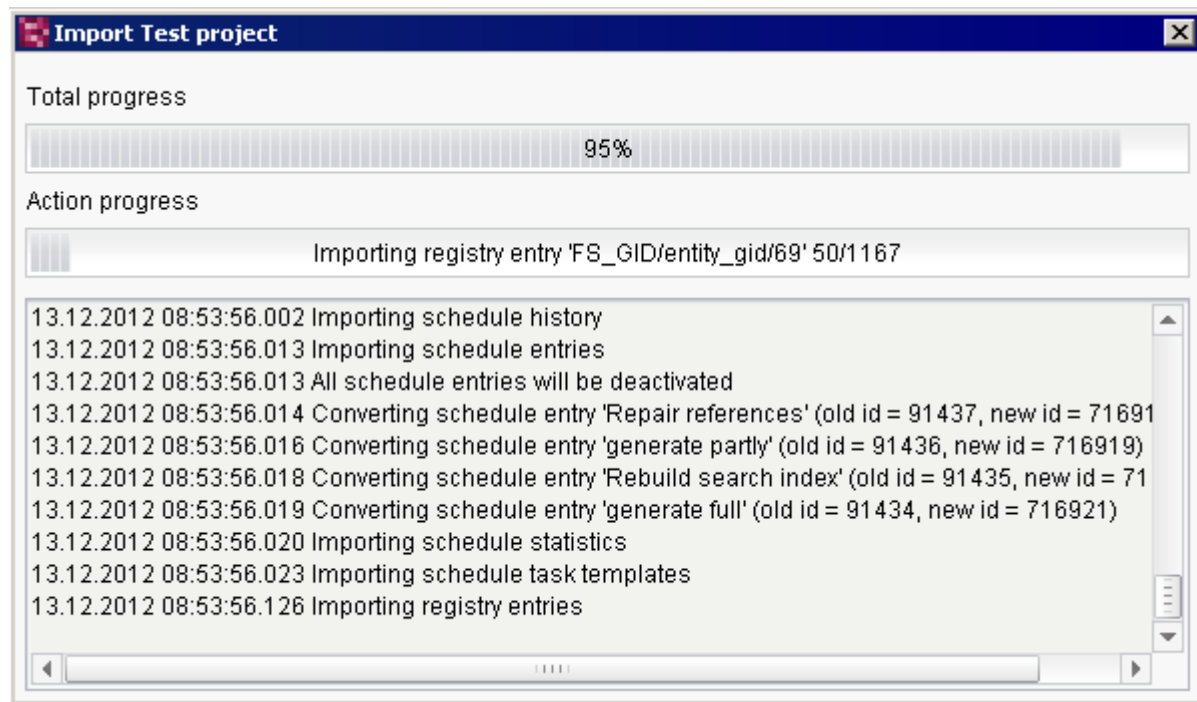


Figure 7-10: Import project – Progress indicator



If errors occur while a project is being imported, importing of the project will not be canceled, but will continue (e.g., in the case of failed database synchronization). At the end of the import process, an error message will appear. The corresponding exceptions will appear in the import log. The project is then deactivated after import.

7.2.3.3 Export

This function is used to create an export file for each project.



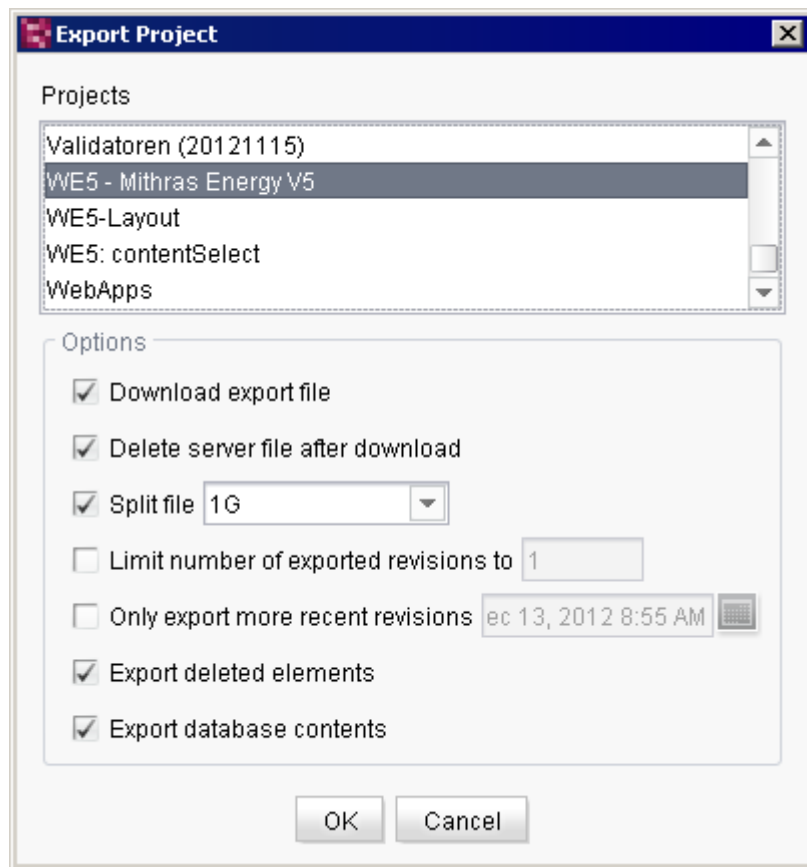


Figure 7-11: Project – Export

All projects currently on the server are listed in the upper section of the dialog box. The project to be exported can be selected from these projects.

Download export file: if this option is selected, the export file will also be saved to the local computer's file system.

After confirming these settings by clicking **OK**, a window appears where the user can select the destination for the export file.

Delete server file after download: this option can only be selected when the export file is downloaded locally. If this option is selected, the export file on the server will be deleted immediately after it is downloaded to the local computer.

Split file: this option specifies a maximum size limit for the export file. Several export files are then created based on the set limit. (This option is required for particularly large projects, for instance, so that they can be copied to external data storage media.)

This option only affects the export file download. In the case of a server-side export, a project that is larger than 1 GB is always split into files 1 GB in size. There is no way to modify the size



of this split.

Limit number of exported revisions to: this specifies the number of revisions created during export. When limiting the revisions to 1, only the current state of the project is exported (see Chapter 9.17 page 515).

Only export more recent revisions: this sets a maximum age limitation for the revisions that are to be exported. If the current date is used for this setting, only the current state of the project will be exported.

Export deleted elements: selecting this option includes deleted elements in the export file. If the option is disabled, all deleted elements are ignored during the export and are no longer available for re-import under the "Restore deleted elements" context menu item.

Export database contents: this option makes it possible to control the export of database content. If this option is selected, all data sources and their content are exported. If the option is disabled, the data sources alone are exported without their content. Only the empty tables are left.



Subsequent changes to the "Allow empty value" option within the FirstSpirit database schema configuration (deletion of the attribute nullable=1, changing from NULLABLE to NOT NULLABLE) are not applied in the database (not even if the Schema sync option is activated). This can lead to problems when exporting and reimporting a project. This situation is checked when a project is exported: If the project to be exported already contains maintained database content with an empty value in a column (NULL), for which the Allow Empty value option was subsequently deactivated (NOT NULLABLE), then the project export is not successfully completed (with corresponding error message).



7.2.3.4 Deactivate

If a project is removed temporarily (but not completely deleted) from the project list, it can be deactivated here.

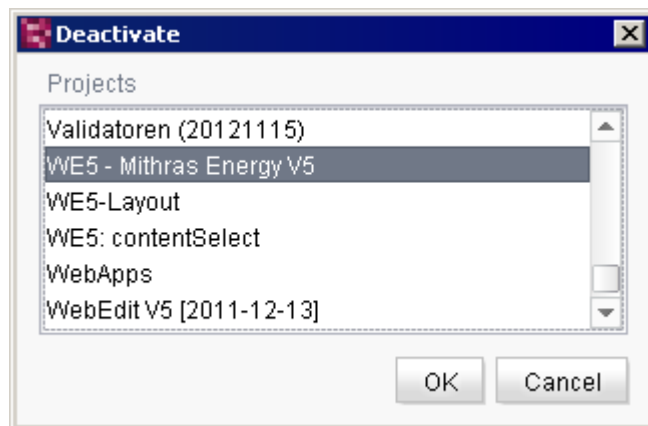


Figure 7-12: Project – Deactivate

Here, the project to be deactivated is selected from the displayed list and its selection is confirmed by clicking "OK". It is possible to select multiple items at once by pressing the SHIFT or CTRL key at the same time. If there are still open sessions on a project, a warning will appear before it is deactivated. The project can still be deactivated.

Deactivating a project is also required before a project can be deleted.

This menu item is only available to server administrators.

7.2.3.5 Reactivate

Here you can re-release deactivated projects for editing. A selection window displays the currently deactivated projects.

This menu item is only available to server administrators.

7.2.3.6 Delete

This function allows users to delete files completely from the server. For reasons of security, however, only those projects that were previously deactivated may be deleted. A picklist displays these projects.



Multiple projects can be selected in the picklist for simultaneous deletion.

A confirmation prompt appears before the projects are deleted.



Deleting a project means that all of the project files are completely removed from the server hard disk. This procedure is irreversible.

This menu item is only available to server administrators.

7.2.3.7 Properties

When this function is called, a window appears where a project must first be selected.

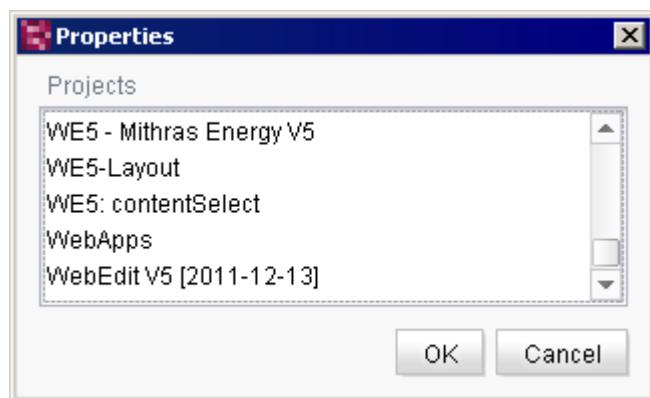
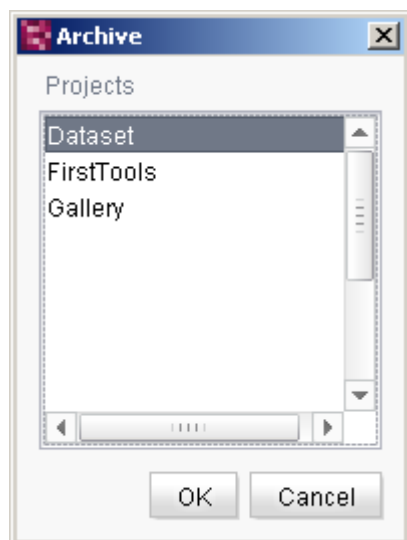


Figure 7-13: Project – Properties

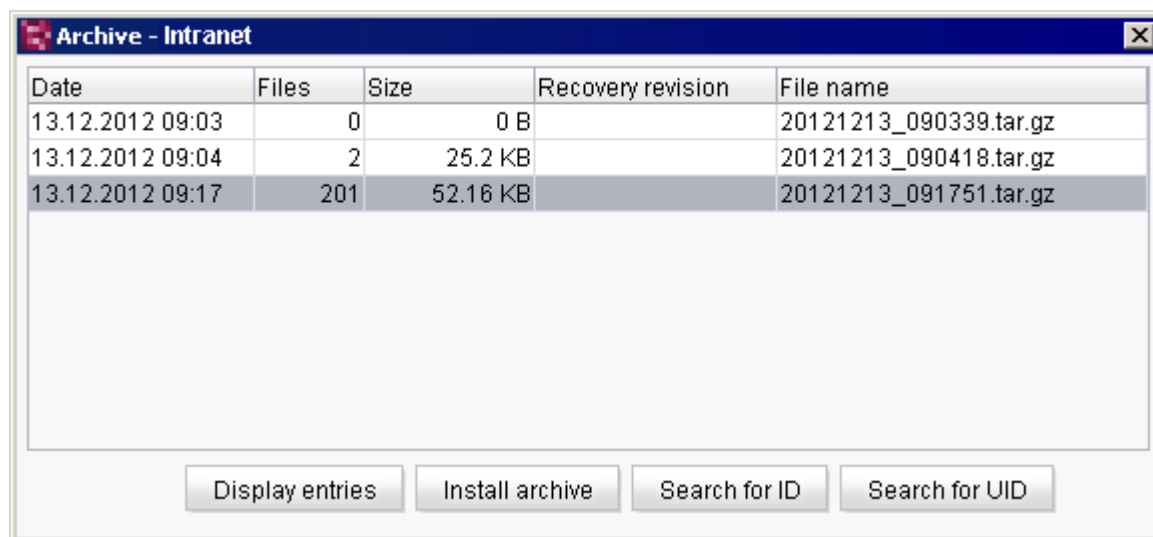
After selecting the project, the project settings dialog appears. For more information, refer to the project settings (Chapter 7.3.13, starting on page 278).



7.2.3.8 Archive

**Figure 7-14: Project – Archive**

The "Archive" function is used to display data from archived files that were created during a scheduled archive (see Chapter 7.5.10.1 page 402):

**Figure 7-15: Archive list**

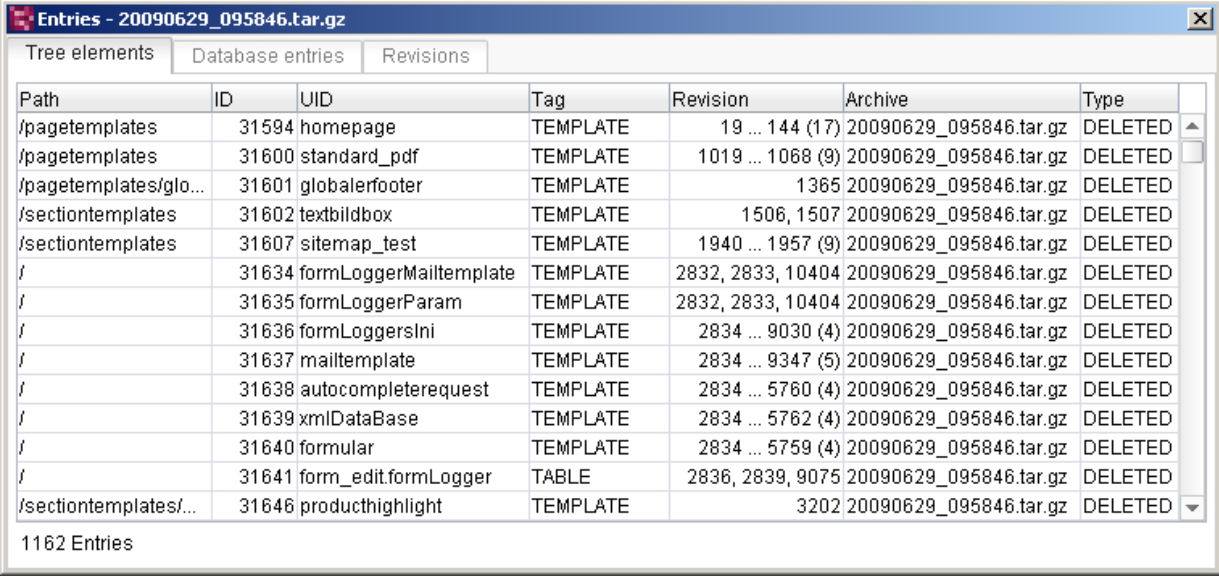
After selecting a project from Figure 7-14, all available archive files are displayed with their creation date, number of files (archived elements), archive file size and file name. Deleted archive files are shown in gray.



Display entries: clicking this button shows all entries of an archive file. They are displayed on three separate tabs covering components, database entries and revision information.

- Components (see Chapter 7.2.3.8.1 page 230)
- Database entries (see Chapter 7.2.3.8.2 page 231)
- Revisions (see Chapter 7.2.3.8.3 page 232)

7.2.3.8.1 Archived tree elements



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	textildbox	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 7-16: Archived tree elements

The following data are displayed in this list for each archived tree element from the FirstSpirit stores:

Path: path to the element where the data have been 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 shown in their entirety. If there are more than three revisions, the first and last revisions are shown and the number of archived revisions is displayed next to them in parentheses.

Archive: name of file in which the element was archived.



Type: shows the reason for archival (for instance, `OLD_VERSION` means that an older and thus obsolete version of the object has been archived, `DELETED` means that a deleted object has been archived).

7.2.3.8.2 Archived database entries

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 7-17: Archived database entries

If database entries were archived using the archive schedule entry (see Chapter 7.5.10.1 page 402), they will appear in this list with the following information:

Schema ID/UID: ID or reference name of the schema from which the archived information related to the respective data record originates.

Table: name of the table from which the archived information related to the respective data record originates

FS_ID: ID of the data record for which the information was archived. Entries with the same FS_ID indicate that changes were made to the respective data record.

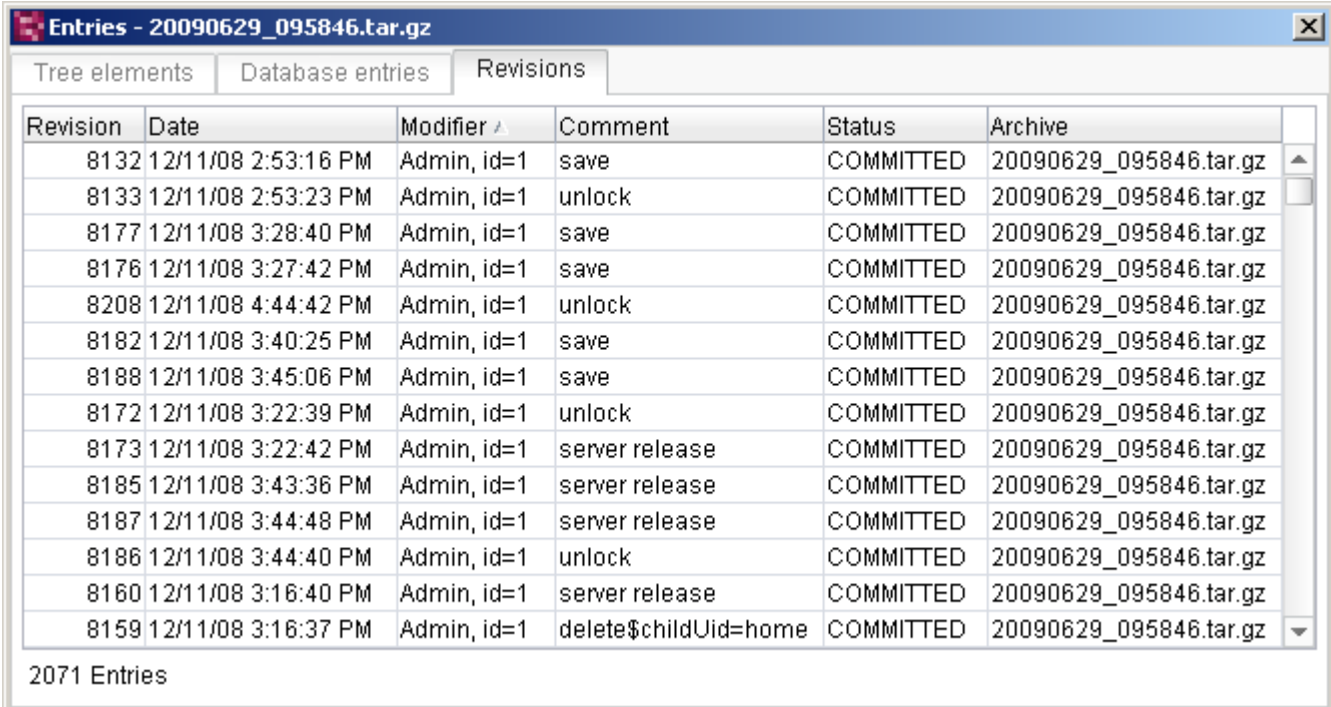
Valid from/Valid to: similar to revisions for tree elements, these data specify the period in which an unchanged version of the data record was present. If changes were made to a data record, the different versions of the data record can be identified based on this space of time.



Release to: specifies the point in time up to which the data record was released.

Archive: name of the file in which the information on the database entry is archived.

7.2.3.8.3 Archived revisions



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 7-18: Archived revisions

If the "System data" option in the objects section of the project archive dialog is selected (see Chapter 7.5.10.1 page 402), the archived revisions are displayed in this list with the following information:

Revision: the archived revision number.

Date: revision date.

Modifier: name of user who made the modification.

Comment: automatically assigned comment.

Status: specifies the editorial status of the revision (e.g. `COMMITTED` for concluded revisions, `UNCOMMITTED` for revisions that have been made, but are not finished)



Archive: name of file in which the revision was archived.

Install archive: this button is used to re-install the selected archive in the project along with all associated elements. The archive file is not deleted as a result of installation and can be reinstalled at a later date.

If you want to restore an archive file after further archiving operations have been performed, you must also install all the other archive files stretching right back to the one you want. These archive files must be installed in reverse order (in reverse date order), i.e. starting with the one that was created last. Otherwise, gaps may be created in the version history.

In the example below (Figure 7-19), it is assumed that the archived element you want to restore is located in the archive file from March 16, 2014:

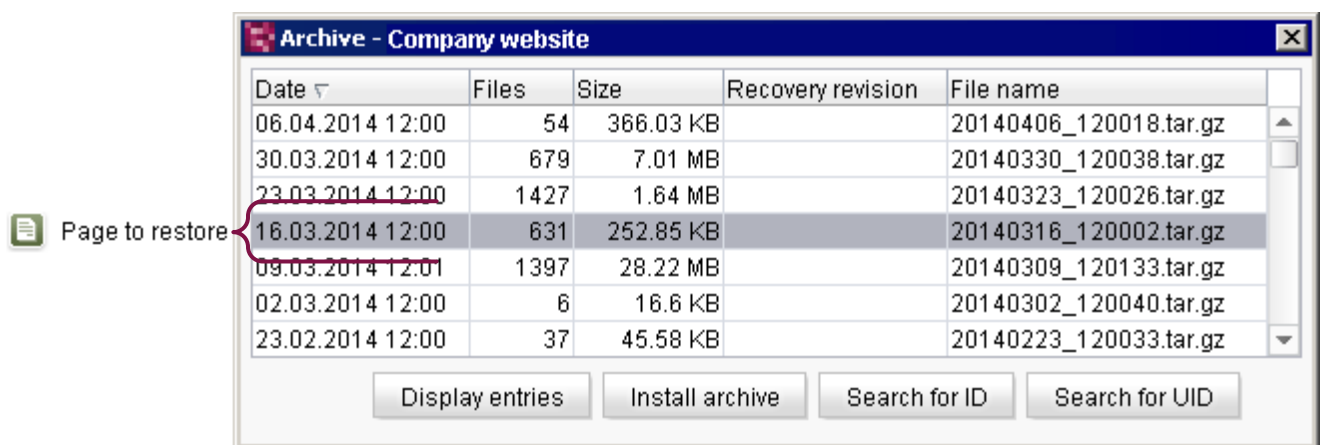


Figure 7-19: Archived element to be restored

In this case, the archive files below must be restored in the following order ("Install archive" button):



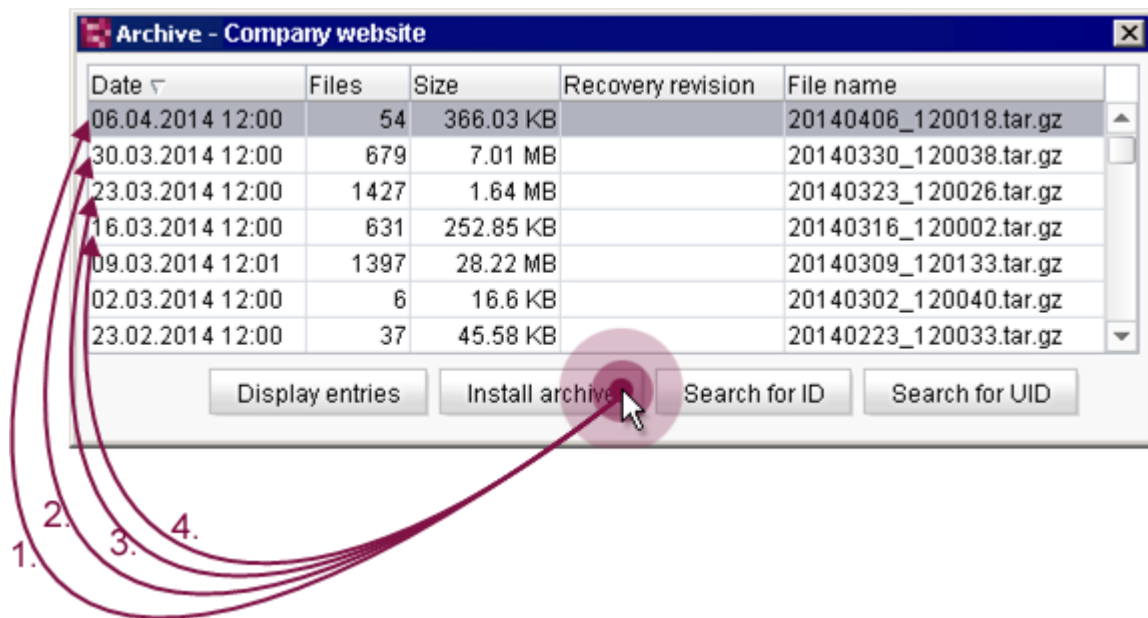


Figure 7-20: Restoring archived elements

1. Installation of archive file dated April 6, 2014, 12 p.m.
2. Installation of archive file dated March 30, 2014, 12 p.m.
3. Installation of archive file dated March 23, 2014, 12 p.m.
4. Installation of archive file dated March 16, 2014, 12:00 p.m.



Deleted archive files cannot be installed.

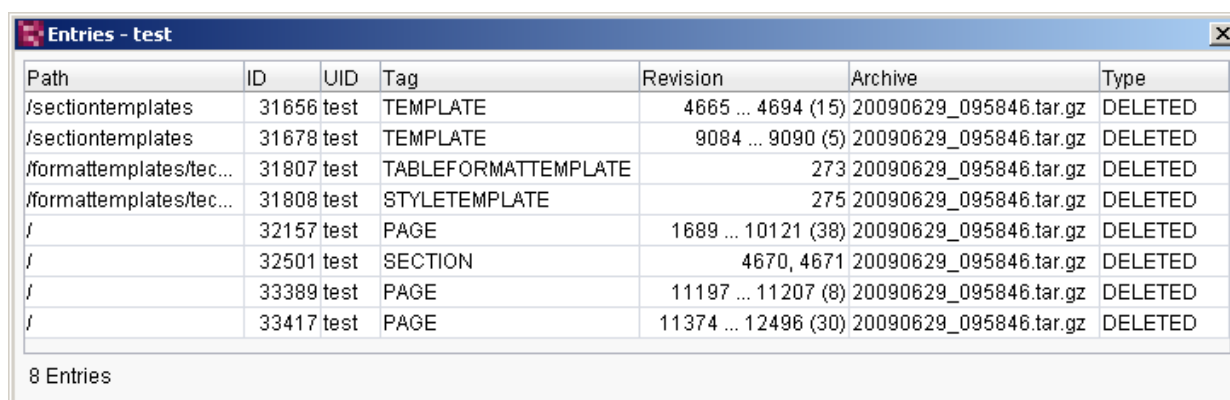
If this order of installation is not followed, "IllegalArgumentException" error messages appearing as "unknown revision id: 18, latest: 46" may occur when calling up the version history for an affected object.

Search for ID: this button is used to search through the archives for IDs of the archived elements. The search runs through all of the project's existing archives.

Search for UID: this button is used to search through the archives for UIDs of the archived elements.

The search runs through all of the project's existing archives. 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

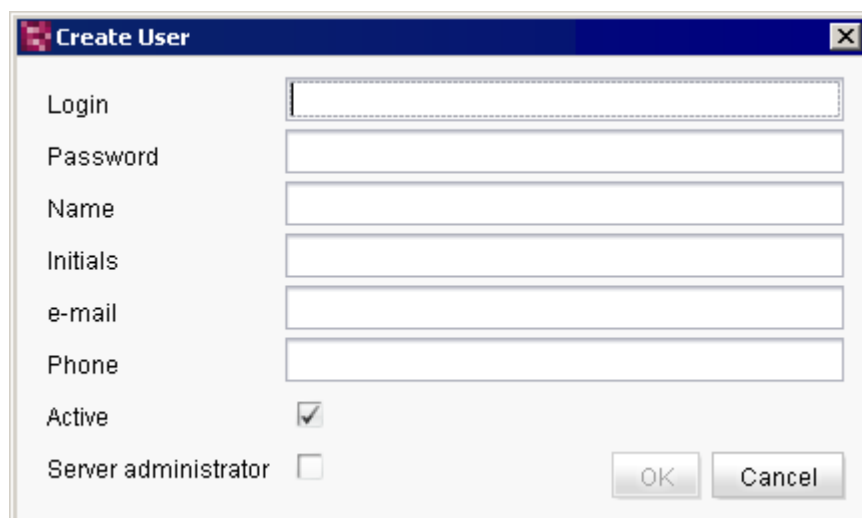
8 Entries

Figure 7-21: List of search results for "deleted" UIDs

7.2.4 User

7.2.4.1 Create User

This function is used to add a new user to the server. A window appears where the user data can be entered.



Create User

Login:

Password:

Name:

Initials:

e-mail:

Phone:

Active: ☒

Server administrator: ☐

OK Cancel

Figure 7-22: Create User

Login: login name of new user (mandatory field).

Password: password of new user (mandatory field).

The other information about the new user, such as the actual **Name**, **Initials**, **e-mail** address and **Phone** are optional.



"External user" exception: if the user is identified through an external system, he is created automatically as a FirstSpirit user the first time he logs in. The required user attributes in this case are imported from the external system (the password is pre-populated in FirstSpirit with a random value). The user then appears in the list of FirstSpirit users (see Figure 7-23).

Active: if this option is not selected, the newly created user is added to the system, but cannot be authenticated (for information on "deactivating users", see Chapter 7.2.4.2 page 237).

Server administrator: Use this option to assign the role of the server administrator to the newly created user. He/she has all permissions

- in ServerManager
- in ServerMonitoring
- in their own connections, set up via API

by default. If the server administrator permissions are to take effect in SiteArchitect, this can be activated via the entry "Administrator mode" in the "Project" menu (see *FirstSpirit Documentation about the SiteArchitect*). If a similar function is required in ContentCreator, this must be implemented using the API (for example method `setAdminMode` (**FirstSpirit Access API**, Interface `User`, Package: `de.espirit.firstspirit.access`, this method can only be executed by server administrators).



If "administrator mode" is activated via API (`setAdminMode(true)`), this does not effect the "Administrator mode" menu item in the "Project" menu of SiteArchitect. The checkmark is not set as a result.

The "server administrator" option is activated for the user "Admin" who is automatically created during the installation of a FirstSpirit Server and cannot be deactivated. It can only be assigned by server administrators; initially, therefore, it can only be assigned by the administrator (user ID 1). If a user who had server administrator permissions on the other FirstSpirit Server is created by importing a project, this permission is removed during the import and must be regranted as necessary.

If the server administration permissions are assigned to a user via ServerManager, this is recorded in the file `fs-server.log`, stating the user name:

```
INFO 02.10.2013 10:43:05.767
(de.espirit.firstspirit.server.usermanagement.UserManagerImpl): Setting user 'chief'
server admin permission to true
```



When a user logs onto the FirstSpirit Server with server administrator permissions, this is also logged accordingly, stating the user name, e.g.

```
INFO 02.10.2013 09:05:21.113
(de.espirit.firstspirit.server.sessionmanagement.SessionManagerImpl): new session
(ID=5030863150308873085, user=chief, userID=62, type=MAIN) created

INFO 02.10.2013 09:05:21.113
(de.espirit.firstspirit.server.sessionmanagement.SessionManagerImpl): Session with
ID=5030863150308873085 bound to ip 192.168.100.212

INFO 02.10.2013 09:05:21.113
(de.espirit.firstspirit.server.sessionmanagement.SessionManagerImpl): User 'chief'
login with server admin permissions, session ID=5030863150308873085
```

External users can be made server administrators via a corresponding LDAP configuration; specifically, via a corresponding parameter in the configuration file `fs-server.conf`. See Chapter 4.3.1.2 page 37, parameter `externalServerAdminGroup`.

For information about the differentiation between administrator, server administrator and project administrator see Chapter 7.1 page 213.

7.2.4.2 Edit

This function is used to edit the above-mentioned user information at a later time. A sorted list of registered users is displayed. (It can be sorted by clicking on any column heading.)

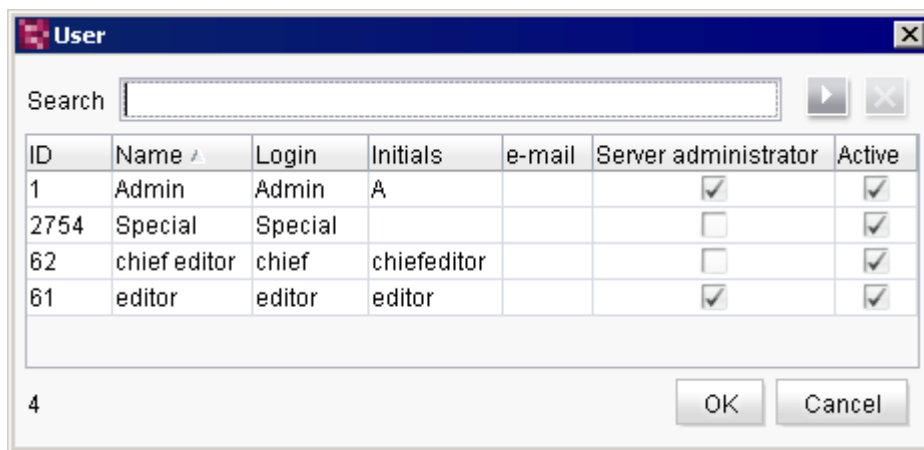



Figure 7-23: Editing a user

Please note that, for **external users with server administrator permissions** this list only reflects the state of the last FirstSpirit login and not the current LDAP state. This means that

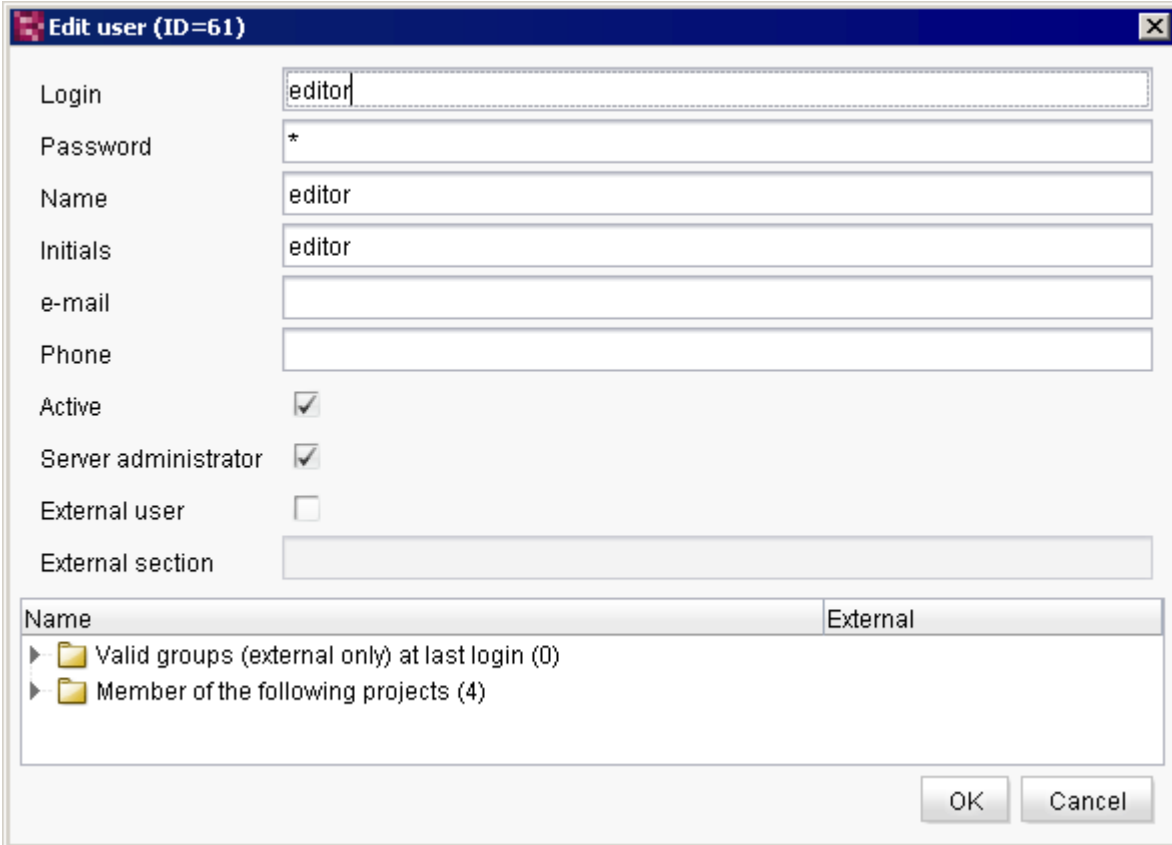
- there may be more server administrators than are marked in the list with a check in the "Server administrator" column, but they have not yet logged onto the FirstSpirit Server via LDAP, or have not done so since authorizations were changed



- a user whose server administrator permissions have been withdrawn in the LDAP and who has not logged onto the FirstSpirit Server via LDAP since will still be shown as a server administrator.

Search: the search function is used to search for words or parts of words in the Name and Login columns. Clicking on  starts the search, and clicking on the button next to it cancels the search.

After selecting the user from the list, the "Edit user" window appears where the user data can be edited.



Edit user (ID=61)

Login: editor

Password: *

Name: editor

Initials: editor

e-mail:

Phone:

Active: ☒

Server administrator: ☒

External user: ☐

External section:

Name: External

- ▶ Valid groups (external only) at last login (0)
- ▶ Member of the following projects (4)

OK Cancel

Figure 7-24: Edit user

If the user was added manually to FirstSpirit, all data, including the user name and password, can be edited. The user is identified internally by a unique user ID. This ensures that user information is preserved (e.g. the assignment to a project) when user attributes (such as the name) are changed.

If the user was created automatically as a FirstSpirit user and is identified through an external system, changes to the user attributes cannot be made in FirstSpirit. External users are



displayed by selecting the "External user" checkbox.

Active: users can be deactivated without having to removing them completely from FirstSpirit. They remain in the system, but they can no longer be authenticated (even in the case of authentication via external systems, e.g. SSO). The deactivated users are grayed out in the FirstSpirit editing environment and in ServerMonitoring. Deactivated users can be reactivated any time in this dialog. All information about the user (such as the original user ID) and all project assignments remain unchanged and can be used again immediately (as opposed to if a user is deleted and then recreated).

Server administrator: Use this option to assign the role a server administrator to this user. See also explanations about the option "Server administrator" in Chapter 7.2.4.1 page 235.

"External user" exception: if the checkbox is *selected*, the user is from an external system (e.g. from LDAP) who has been added to the server automatically (see Chapter 7.4.7.2 page 314). If the checkbox is *unchecked*, the user was added manually (see Figure 7-22).

External section: the LDAP section where the user is registered is displayed here. If the user logs in via the WindowsLoginModule, the domain is displayed here as the external section.

Group membership: the user's group membership is displayed at the bottom of the dialog box. The assignment is subdivided by projects and cannot be edited in this dialog box. For information on changing a user's group membership, see Chapter 7.4.8.5 page 320. This information is also available in FirstSpirit™ ServerMonitoring (see Chapter 8.5 page 469).

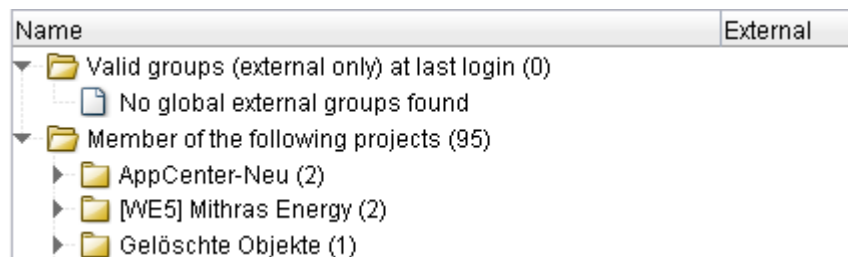


Figure 7-25: User's group membership

The other information about the new user, such as the actual **Name**, **Initials**, **e-mail** address and **Phone** are optional. In the case of external users, these fields may be populated automatically using user attributes.

The internal system user ID is assigned automatically and cannot be changed. Permissions for all users can be managed using the SiteArchitect context menu.

This menu item is only available to server administrators.



7.2.4.3 Delete

This function is used to remove a user from the server.

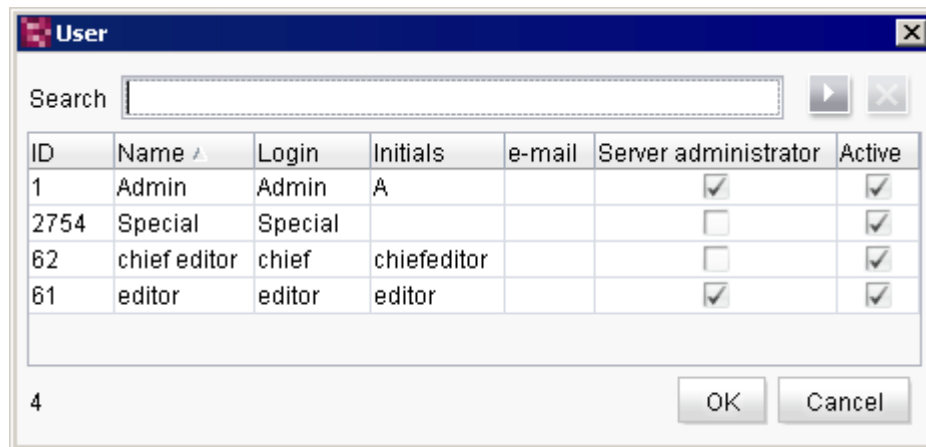



Figure 7-26: Deleting a user

Search: the search function is used to search for words or parts of words in the Name and Login columns. The search is started by clicking on the  button.

After selecting the user from the list, the user is deleted from the list once the deletion prompt is confirmed. A distinction is made here between the following:

- a. Manually added FirstSpirit users:
If the user was added manually, the user is automatically removed from all FirstSpirit projects and deleted from the server using the "Delete user" function.
- b. Automatically added (external) users:
If it is an external user, the user is automatically removed from all FirstSpirit projects and deleted from the server using the "Delete user" function. However, if the user has not been deleted from the external system, the user will be added again as a new FirstSpirit user with a new user ID the next time the user logs in.

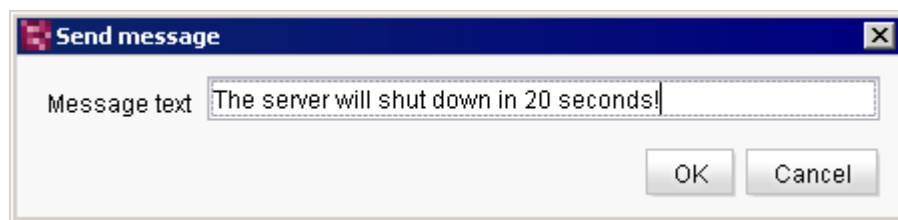
This menu item is only available to server administrators.



7.2.5 Extras

7.2.5.1 Send message

The text of a pop-up window message that will be sent to all active clients on the server can be edited here.



This is particularly useful when the server is to be shut down. The editors are therefore given sufficient time to save their work so that they do not lose any data.



7.2.6 Help

7.2.6.1 About FirstSpirit

This function provides an overview of some of the information about the version of FirstSpirit that is currently being used:

FirstSpirit ServerManager 5.1.123.45678

Server: localhost:8000 (HTTP)

Project:

User: Admin

Groups (External):

Server version: 5.1.123.45678

Licensed to: e-spirit

Memory: 68.63 of 494.94 Mbyte occupied

Java version: 1.7.0_45 32bit Oracle Corporation

Operating system: Windows 7 6.1 x86

Project loading time: 0.00 s 0.00 kbyte/s

FirstSpirit™
Your Content Integration Platform



7.2.6.2 Index

Selecting this function opens the "Manual for Administrators" (.pdf) in the browser. The "ServerManager" chapter is displayed.



7.3 Server properties

The "Properties" menu item under the "Server" menu item can be used to configure the properties of FirstSpirit Server.

7.3.1 Global server properties

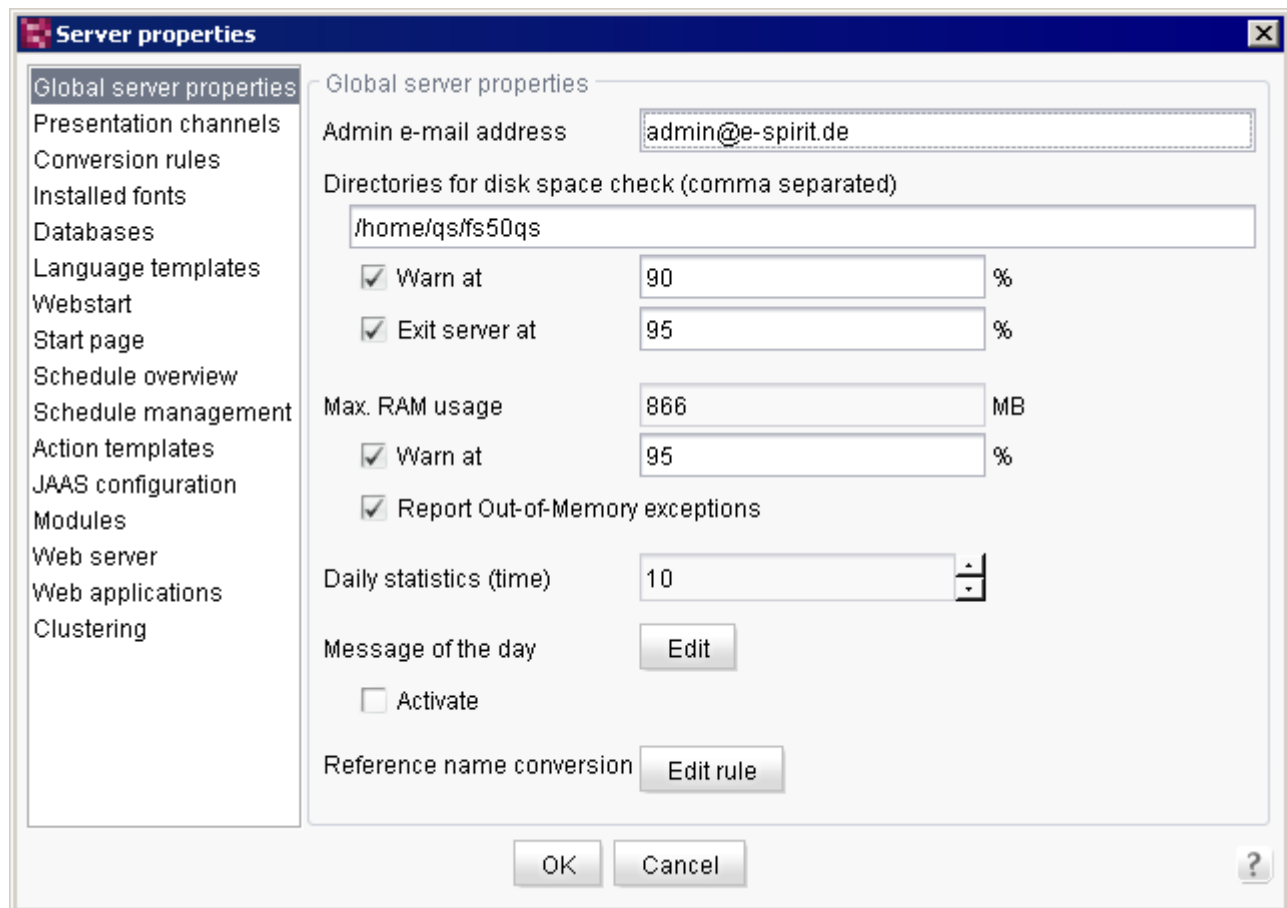


Figure 7-27: Server properties – Global server properties

The first menu item called "Global server properties" covers the following configuration options:

Admin e-mail address: enter the e-mail address of the responsible server administrator in this field. Critical errors that affect the configured fields following this one can be sent by e-mail to the e-mail address specified here. If no e-mail address is specified, the relevant error messages will be output to the server's log output only.



Directories for disk space check (comma separated): enter the directories in this field whose remaining disk space is to be monitored. By default, the FirstSpirit Server root directory is entered here (also see `fs-server.conf`, `hdd.directories` parameter, Chapter 4.3.1.15 page 62). If FirstSpirit is distributed across multiple mount points, all mount points of the file systems must be specified that are used by FirstSpirit. Multiple directories must be specified using commas to separate them. The free space in the specified directories is determined every five minutes.

Warn at / Exit server at: this field specifies at what percentage the warning e-mail is to be sent to the server administrator and/or at what percentage the FirstSpirit Server, including relevant messaging, is to be shut down.

Warning e-mail message:

"FIRSTspirit server 'MYSERVER' disk space warning: /home/fs/firstspirit5, 32,01 GB free, 42,52 GB used".

FirstSpirit Server shutdown e-mail message:

"FIRSTspirit server 'MYSERVER' disk space shutdown limit reached: /home/fs/firstspirit5, 463.2 MB free, 5.54 GB used".

The percentage is the ratio of the disk space available in the specified directory to the space already filled. By default, the value is set to 90% for the warning and 95% for FirstSpirit Server shutdown. This means that the server administrator receives a warning e-mail when the disk is 90% full (see also `fs-server.conf`, `hdd.limit` parameter, Chapter 4.3.1.15 page 62) and the server is shut down when the disk is 95% full (see also `fs-server.conf`, `hdd.shutdown` parameter, Chapter 4.3.1.15 page 62). If multiple directories are specified, an e-mail is sent or the server is shut down when the corresponding percentage is exceeded in **one** of the directories. The value selected for a warning should be less than the value for the "Server shutdown" property. A warning e-mail is sent no more than every 12 hours.

The respective checkbox enables or disables the sending of a warning e-mail (see also `fs-server.conf`, `hdd.limit.active` parameter, Chapter 4.3.1.15 page 62) or server shutdown (see also `fs-server.conf`, `hdd.shutdown.active` parameter, Chapter 4.3.1.15 page 62). The two checkboxes are selected by default.

Max. RAM usage: the maximum RAM usage (in MB) can be set for the server VM in this field. This value is passed when the server is started. A critical threshold (in %) can be defined as well. If the adjacent checkbox is selected, a warning is written to the server log as soon as the critical RAM usage threshold is exceeded (an e-mail can optionally be sent to the server administrator). If the "Report Out-of-Memory exceptions" checkbox is also selected, an e-mail is sent to the server administrator when an OutOfMemory 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 ServerManager "Max. RAM usage" field:

- 90% (HEAP STORAGE used / HEAP STORAGE max=91%, which is more than 90%)

Daily statistics (time): a starting time for running the daily statistics can be set in this field.

Message of the day: clicking the **Edit** button allows the user to specify text for each project language that will appear as a message during startup. In FirstSpirit SiteArchitect, the message initially appears when a project is opened if the "Activate" checkbox is selected.

Reference name conversion: this function is used to define a server-wide set of rules used for the conversion of invalid characters when creating new FirstSpirit objects or when modifying the reference name. Clicking on the "Edit rule" button opens the following window:

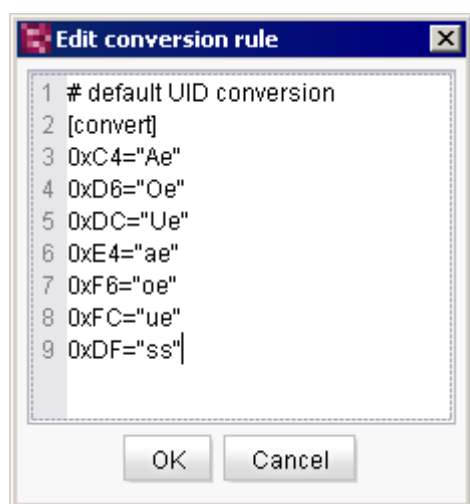


Figure 7-28: Edit conversion rules (converting reference names)



This window by default already contains some defined rules for the upper and lower case umlauts "ä", "ö" and "ü" and for the "ß" character. Each rule must be on its own line and consists of two values separated by an equal sign:

- **To the left of the equal sign:** the special character to be converted using the ASCII (hexadecimal) code.
- **To the right of the equal sign:** the valid character(s), placed within double quotation marks, to which the special character is to be converted when used in reference names.

The default set of rules converts umlauts into two characters (vowels to lower case + "e" and "ß" to "ss") when used in reference names



If the characters in this dialog are not coded or formatted correctly, an error message will appear when attempting to save:

"There is an error in the conversion rule format: Error parsing line 11:...".



Only one rule can be defined for a single character and not for strings, and the rule is language-independent only, i.e. each rule applies to all languages. If more than one definition is specified for a special character (ASCII code), the lowest definition in the list will be applied. Special characters without a saved rule are removed immediately after a reference name is entered.



When defining conversion rules for the . (dot) symbol, it is important to note that this is used in FirstSpirit to generate reference names for table templates and is also converted for the corresponding rule definition.

The rules provided can be modified or deleted and new rules can be added. They are not reset or modified during a FirstSpirit update. Comments can be entered using a leading #.

Clicking "OK" saves the rules. If the dialog behind it showing the server properties is closed when "OK" is clicked, the rules are applied immediately. The server/project does not have to be restarted.



7.3.2 Presentation channels

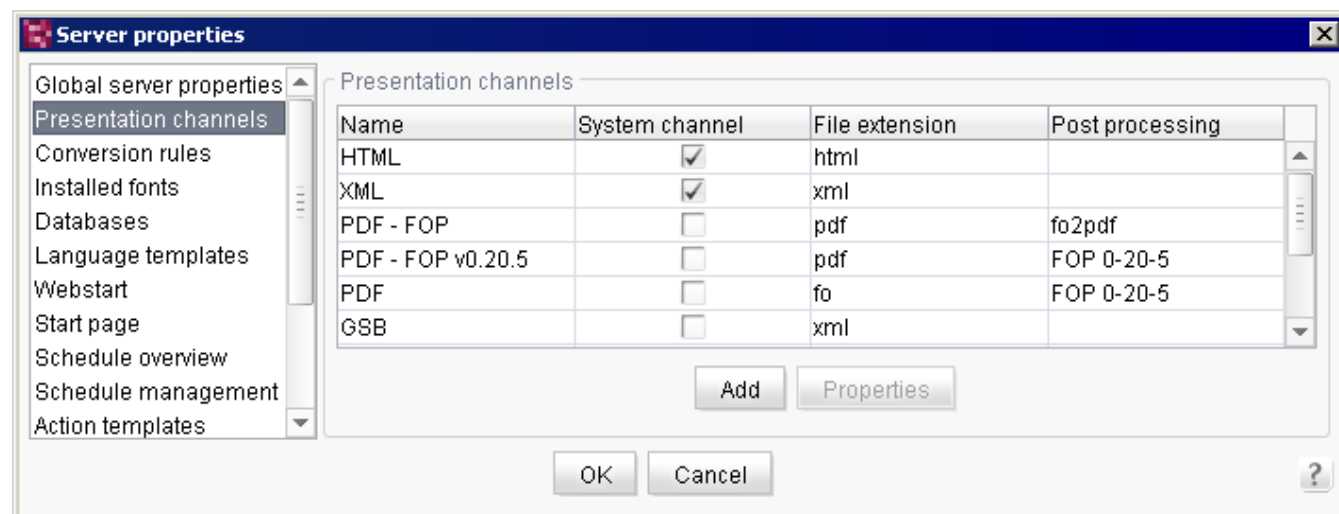


Figure 7-29: Server properties – Presentation channels

This is where the server presentation channels are defined. The template sets found on the server are based on the presentation channels defined here.

Add: clicking on "Add" opens a dialog box where a new presentation channel can be defined (see Figure 7-30).

Properties: the "Properties" button is active if a presentation channel is highlighted in the list. Clicking on this button opens a dialog box where the user can edit the presentation channel properties (see Figure 7-30).



Figure 7-30: Server properties – Edit presentation channel

Name: name of the presentation channel. This name is displayed for the presentation channel in the project and within the list of presentation channels after it is saved.



System channel: this checkbox indicates if the presentation channel is a system channel. These presentation channels cannot be edited or deleted.

File extension: the presentation channel file extension is specified here; e.g. "html" for the HTML presentation channel.

Post processing: this option lets the user specify the FOP processor for post processing.

For information on support for Apache FOP, see Chapter 7.8 page 441.

7.3.3 Conversion rules

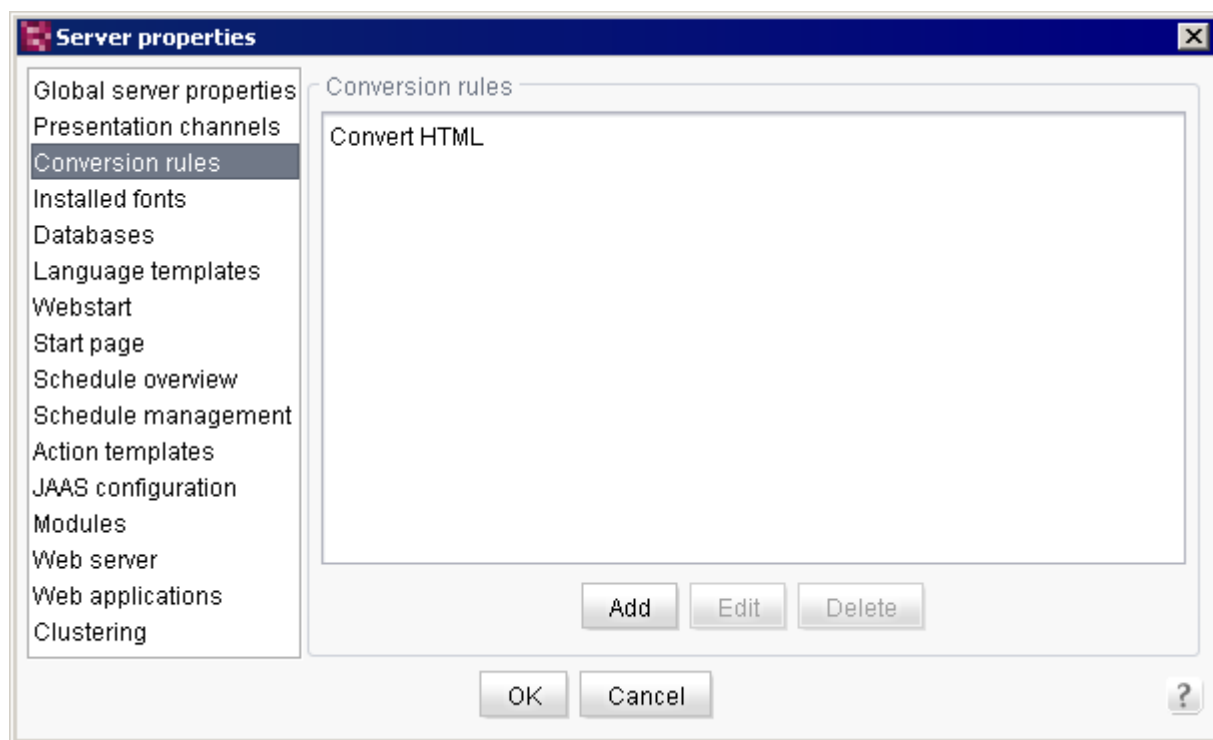


Figure 7-31: Server properties – Conversion rules

This is where the server conversion rules are defined. The conversion rules defined here can be selected from a drop-down list in the project's format templates for each presentation channel. The "Convert HTML" conversion rule is provided by default with the following content:



```
[convert]
0x3c="<"
0x3e=">"
0x22="&#34;"
0x26="&"
0x27="&#39;"
```

Only one conversion rule can be specified for each presentation channel.

Conversion rules are used to convert characters that are input. 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" should also be evaluated, the relevant option in the format template needs to be selected.

For more information, see FirstSpirit Online Documentation.

Add: a selection dialog opens in which the local computer file structure can be searched. A new conversion rule needs to have been added in advance as a text file.

Edit: the "Edit conversion rule" dialog is used to edit an existing conversion rule.

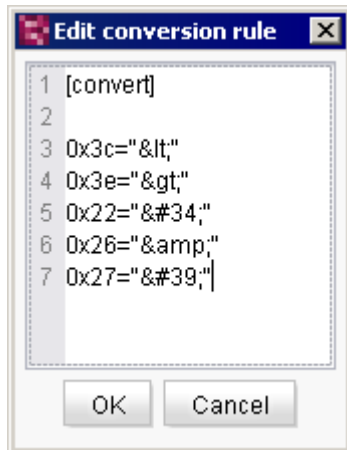


Figure 7-32: Server properties – Edit conversion rule

Delete: clicking on this button lets the user remove existing conversion rules from the server. A confirmation prompt appears before deletion.



7.3.4 Installed fonts

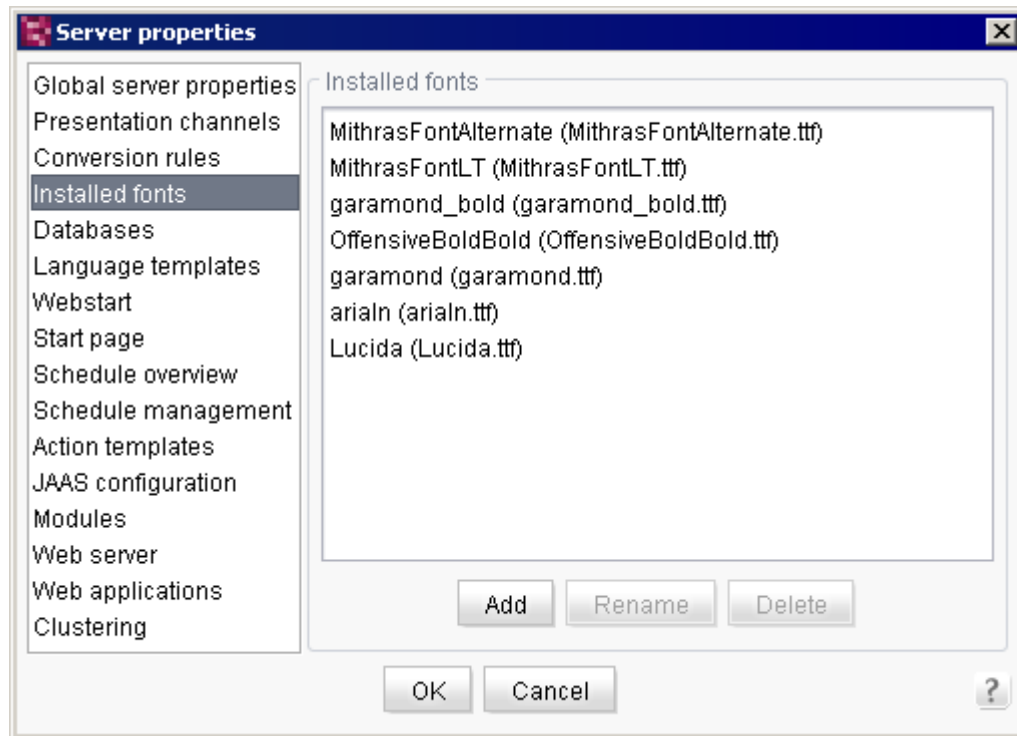


Figure 7-33: Server properties – Installed fonts

This provides a list of the True Type Fonts (TTF) installed on the server. These fonts can be accessed within a template in order to label photos, for instance, with text in the corresponding font:

```
$CMS_REF(cmsFont(font:"MetaMediumCaps",size:12,color:"#666666",justify:"right",valign:"center",xoffset:-4,yoffset:-2,media:"version_from",bounds:"image",text:#content.toString))$
```

It shows the reference name used to reference the font and the file name in parentheses used to store the font in the server file system (see Figure 7-33).

Add: if another font is to be added, it must already be installed on the server computer operating system. It is not enough to install the font on the client computer operating system. To add a new font to the server, a unique reference name for the new font must first be specified. Clicking on "Font upload" brings up the option to select a True Type Font from the file system.



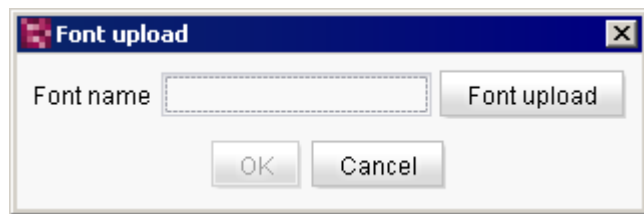


Figure 7-34: Font upload

Clicking on "OK" adds the new font.

Rename: renames one of the fonts shown in the list.

Delete: deletes the highlighted font in the list. The font is removed from the server directory `data/fonts` and no longer appears in the list of installed fonts.



Renaming or deleting a font may cause errors in templates that use this font. A security warning will therefore appear.

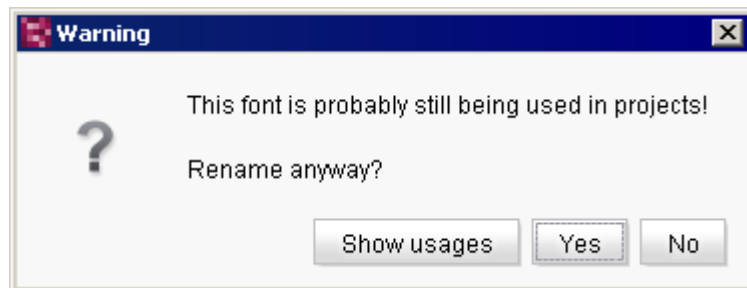


Figure 7-35: Warning before deleting or renaming fonts

Clicking on "Yes" will rename or delete the font. Errors may occur in projects that use this font. In case of doubt, these projects can be found using the "Show usages" button.

Clicking on "Show usages" shows where a font is used within projects. The projects are displayed in a list. A message is displayed if no messages were found.

Clicking on "No" closes the dialog. The affected font is not deleted or removed.



7.3.5 Databases

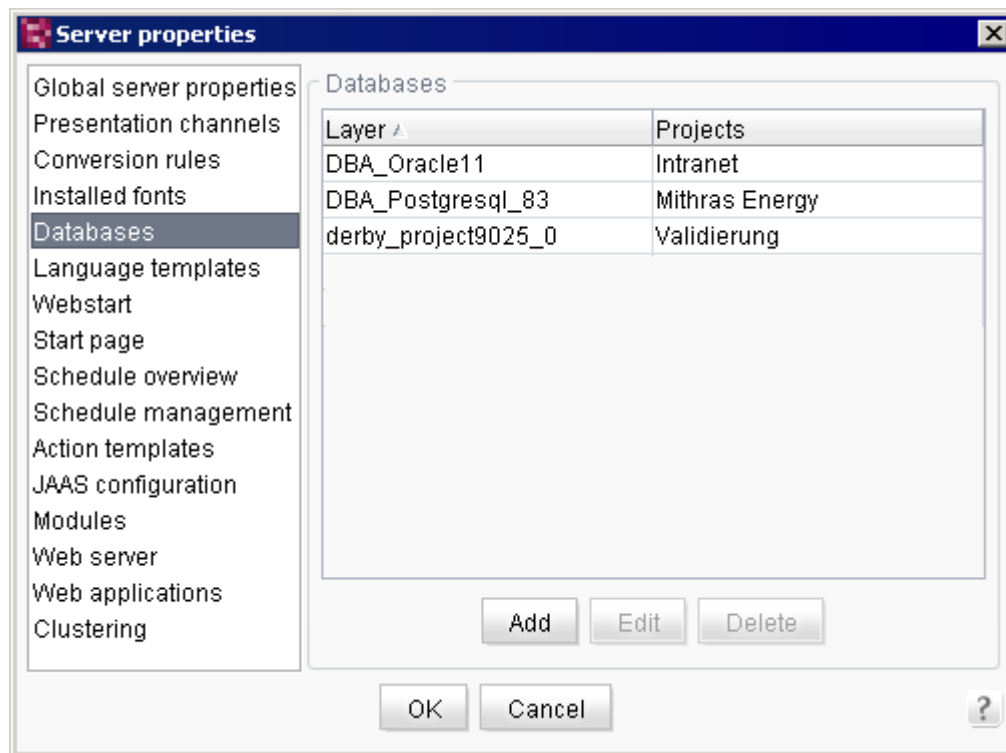


Figure 7-36: Server properties – Databases

All connected database layers on the server, i.e. all connections to a database on the server, are listed here. A database layer defined here is selected within a project's properties (see Chapter 7.4.12 page 325) and can then be selected in the project's template store when defining a database schema (for more information on using database layers in a project, see the *FirstSpirit Manual for Developers (Basics)*).

Clicking on the "Add" button lets the user add a new database layer to the server. A window opens in which only the name of the new configuration needs to be entered.



Figure 7-37: Creating a new database layer



A window then appears where the database can be configured. The basic structure for a MySQL database is already present. The configuration of the database must then be adapted to the local requirements (see Chapter 7.3.5.1 and the following pages). (For more information on binding databases, see Chapter 4.3.3 page 84, and on configuring databases, see Chapter 4.9.1 page 155.)



FirstSpirit uses JDBC²⁹ to access databases. For each database a suitable driver must be included in the server class path or the JDBC driver files need to be integrated as a FirstSpirit module (see also 4.9.1 page 155).

An existing configuration can be edited by clicking on the "Edit" button. A window containing the existing configuration opens (see Figure 7-38). A distinction is made here between the following:

- JDBC parameter configuration (see Chapter 7.3.5.1 page 254)
- Connection configuration (see Chapter 7.3.5.2 page 255)

Clicking "Delete" removes the existing database layer from FirstSpirit Server. If the layer represents an embedded Derby database, the associated database is deleted in addition to the layer by clicking the button. (For all other layers, the associated database must be deleted manually by the database administrator).



In the case of databases that are currently used in a project, a security warning appears before deletion.

²⁹ Java Database Connectivity



7.3.5.1 Configure JDBC parameters

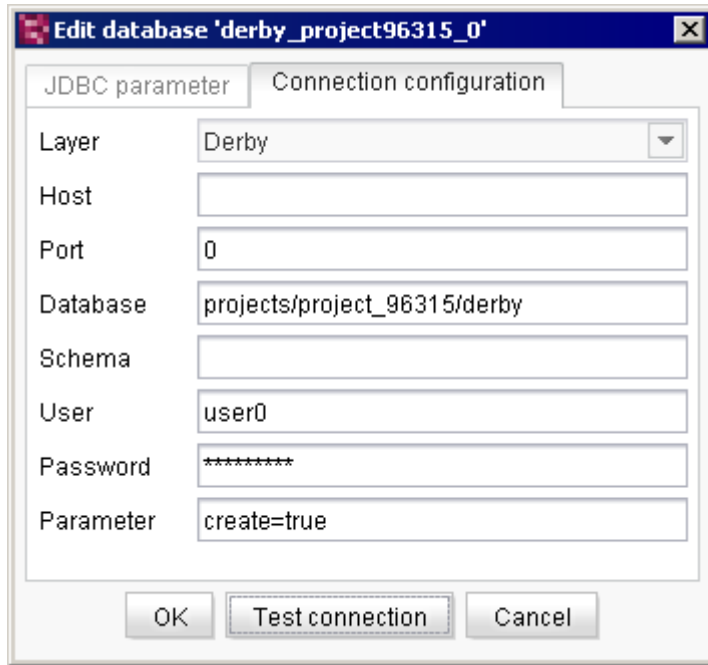
**Figure 7-38: Database connection configuration (JDBC)**

Configuration of the JDBC parameters for the database connection.

The configuration can be changed and then saved by clicking "OK". The connection to the database should be tested before saving changes. For additional information on configuring the database, see Chapter 4.9.1 page 155.

Test connection: after completing the database configuration, the user can use this button to check if all the information was entered correctly.

7.3.5.2 Testing the connection configuration

**Figure 7-39: Database connection configuration (connection)**

Configuration of the connection parameters for the database connection.

Layer: the combo box is used to specify the class that implements the database layer for this specific database system (see the `layerclass`, parameter, Chapter 4.9.4.1 page 169).

Host: host name for the database server bind address. When specifying a Derby layer, this field should remain empty.

Port: port number for the database server bind address. When specifying a Derby layer, this field should remain empty.

Database: name of the database in which the data are to be stored. By default, this value is specified using the name of the database layer.

Schema: there are two types of schema:

- **Standard layer:** an existing DB schema is specified here using the `jdbcc.SCHEMA` parameter in the layer configuration. The name of the schema must be identical to name of the schema within the database. In this case, all database content is written from FirstSpirit to this schema.



- **DBA layer:** the `jdbc.SCHEMA` parameter is not defined here. In this case, every FirstSpirit schema creates its own database schema for the respective database content.

User: valid login name of a database user. This account is used by FirstSpirit Server to establish a connection to the database at runtime (see the `jdbc.user` parameter in Chapter 4.9.4.1 page 169).

Password: valid login password under `<database>.jdbc.USER` (see the `jdbc.password` parameter in Chapter 4.9.4.1 page 169).

Parameters: input option for additional database-specific parameters.

The configuration can be changed and then saved by clicking "OK". The connection to the database should be tested before saving changes. For additional information on configuring the database, see Chapter 4.9.1 page 155.

Test connection: after completing the configuration for the new database, the user can use this button to check if all the information was entered correctly.



7.3.6 Language templates

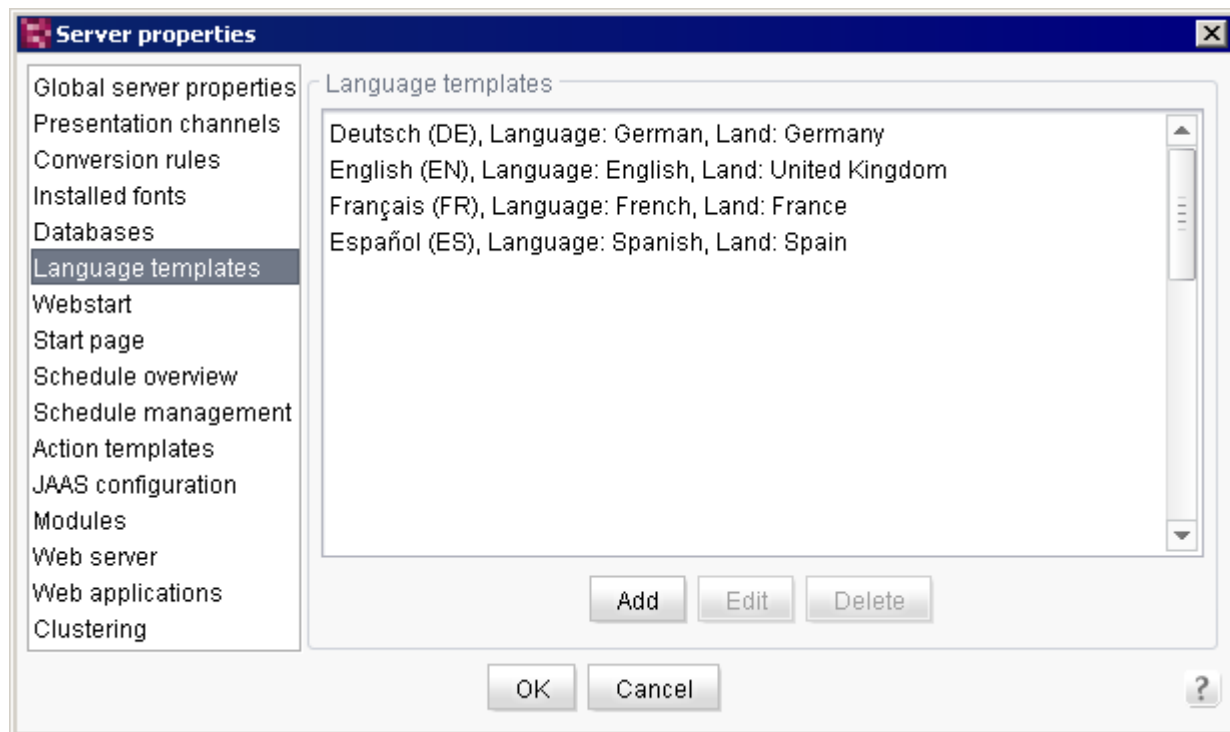


Figure 7-40: Server properties – Language templates

FirstSpirit supports multilingual output of a publication. If a new language is required in a FirstSpirit project, the new language first needs to be added to the server.

Add: this button is used to integrate a language into the system. Clicking on it opens the "New language" dialog box (see Figure 7-41). After a new language is added and defined, all projects can use it, but it needs to be embedded in the particular project before the project can use it (see Chapter 7.4.5 page 307). The language tab in the editing environment will appear and can be populated with content.

Edit: use this button to edit an existing language (see Figure 7-41).

Delete: use this button to delete an existing language.

Editing or deleting a language template will not cause you to lose any content already entered into a project in this language. The content will be preserved as long as the language remains in the project's properties (see Chapter 7.4.5 page 307), since these properties use a copy of the language template. After deleting a language template in the server properties, however, it will not be possible to add the particular language to any project's properties.



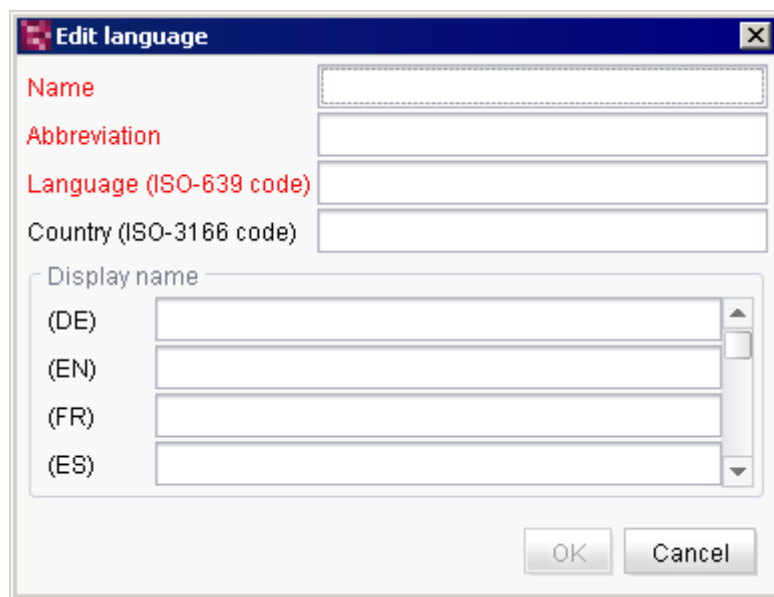


Figure 7-41: Server properties – Adding a new language

The red text indicates a mandatory field. The text will change to black once a valid entry has been made.

Name: a new name for the language must be entered in this field. The name of the language may contain any characters desired.

Abbreviation: an abbreviated name for the language must be defined in this field. The language is displayed in the language tabs within the FirstSpirit editing environment using this identifier. The value specified here is also used in templates for the *lang* attribute (see FirstSpirit Online Documentation, *lang* attribute in all input components, *Template development / Forms* area). The following characters can be used as often as desired: -, _, 0-9 and A-Z. The language abbreviation can be based on the ISO-3166 or ISO-639 standard. In the case of languages spoken in multiple countries, e.g. British English and American English, compound abbreviations are already available, e.g. *EN-GB* or *EN_GB* for British and *EN-US* or *EN_US* for American English.

The language and (optional) country values determine the subsequent language region:

Language: the two-letter ISO-639 based code for the new language must be specified in this field, e.g. *de* for German or *en* for English. Only lowercase letters may be entered. (The first part of the LOCALE)

Country: the two-letter ISO-3166 based code for the country of the new language must be entered in this field, e.g. *DE* for Germany or *GB* for Great Britain. Only uppercase letters may be



entered. (The second part of the LOCALE)

This makes it possible, for instance, to separate en_US (English language with the US as the region; American) from en_GB (English language with the UK as the region; British).

Language-dependent display names can be defined for each language template. The relevant input fields are displayed in the "Display name" area of the form. Language-dependent display names can be defined for all editorial languages of a project.



7.3.7 Webstart

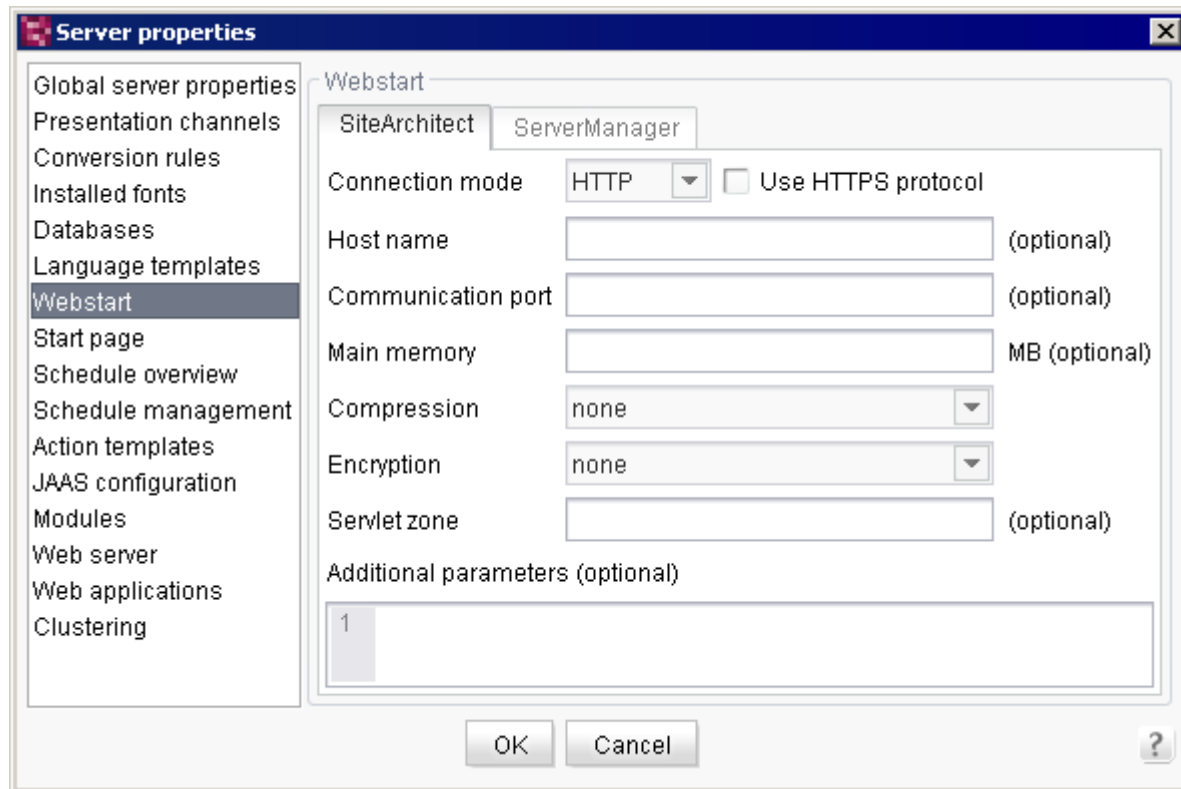


Figure 7-42: Server properties – Configuring Webstart (Quick start)

The quick-start entries in FirstSpirit are configured in JNLP files. The standard configuration (standard JNLP files) can be defined via the "Webstart" menu item.

SiteArchitect tab: configuration for starting SiteArchitect and the quick-start entries on the start page. The parameters configured here affect all quick-start entries on the start page that are of the "Java" SiteArchitect type if no other parameters were explicitly defined for the entry in the "Start page" area.

Individual quick-start entries can be configured via the "Start page" dialog. The generally applicable values are overwritten in the "Webstart" area (see Chapter 7.3.8 page 261).

The configuration options correspond to the user-specific Webstart configuration (see Chapter 6.3.3.1 page 200).

ServerManager tab: configuration for the ServerManager start page. The parameters are similar to those of the quick-start entries on the "SiteArchitect" tab.



7.3.8 Start page

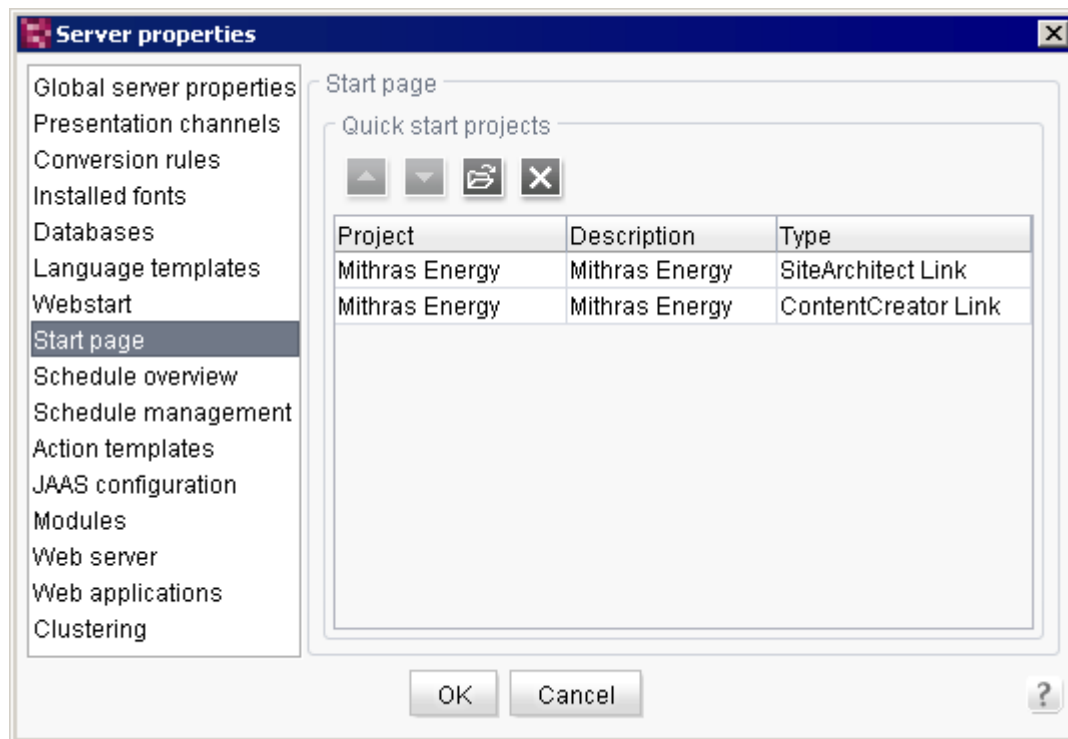


Figure 7-43: Server properties – Configuring the start page

Quick start projects: this selection icon is used to select projects from all projects on the server. The selected projects are then displayed in a table overview and appear on the FirstSpirit start page below the quick-start entries (see Chapter 6.3.2 page 198). Clicking on the Delete icon deletes the highlighted projects in the table. The entries can be sorted using the arrow buttons.

Clicking on a "quick-start entry" on the start page opens the project referenced here directly without the user having to select anything in the project selection list. Depending on the configuration, both SiteArchitect and ContentCreator can be referenced for a project (only projects with a ContentCreator configuration).



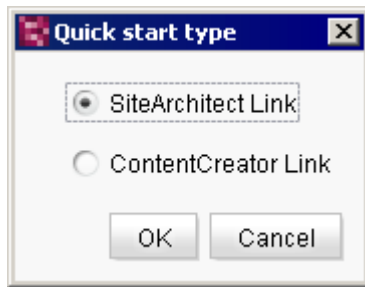


Figure 7-44: Server properties – Adding quick start references

Depending on the setting selected, the selected project is opened in SiteArchitect or ContentCreator. The quick-start entries are labeled "JAVA" or "WEB" to make it easier to distinguish between them.

Additional connection settings (settings for Java Web Start) can be defined for quick-start entries that open in SiteArchitect. Double-clicking on the relevant quick-start entry opens the "Web Start Settings" dialog (see Figure 7-45). The configuration options correspond to the user-specific Webstart configuration (see Chapter 6.3.3.1 page 200).

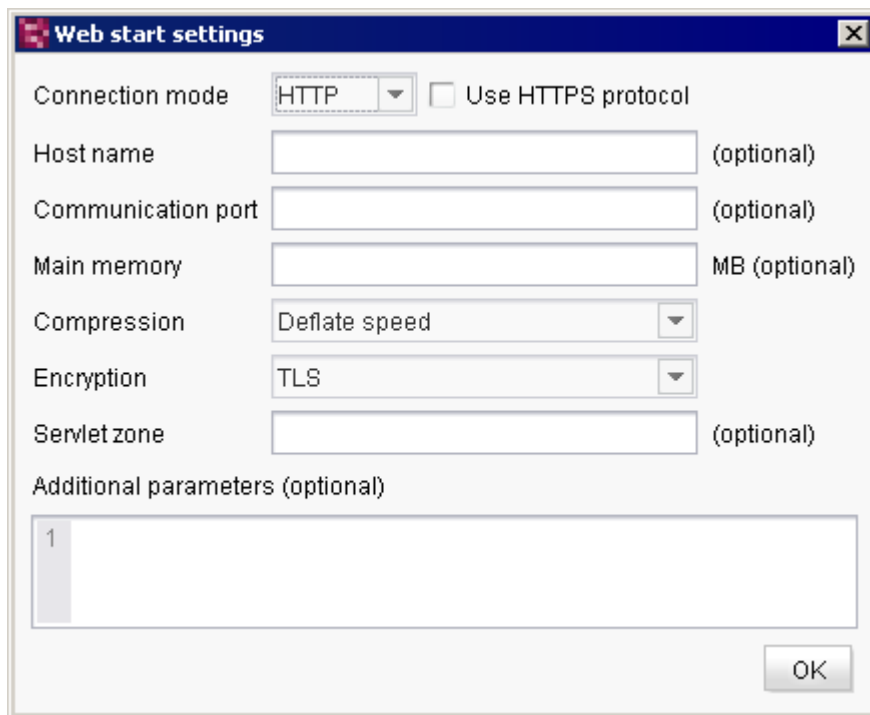


Figure 7-45: Web Start settings for project links (SiteArchitect only)

Evaluation order: the initial configuration is made in the `fs-server.conf` configuration file (see Chapter 4.3.1.1 page 33). The parameters defined there can be overwritten using ServerManager for all Java Web Start applications at once (via the JNLP files) via the "Webstart"



menu item (see Chapter 7.3.7 page 260) and for individual quick-start entries in the "Start page" properties. The evaluation order is:

1. Quick start properties
2. JNLP files
3. fs-server.conf

If the connection settings are defined and activated on the start page (see Chapter 6.3.3.1 page 200), they will be evaluated first.

When deactivating connection settings, the `fs.url` parameters of the `fs-server.conf` file (starting with "`fs.url.`", see Chapter 4.3.1.1 page 33) are not overwritten by the corresponding parameters that are defined in the "Webstart" and "Start page" areas of the server properties.

7.3.9 Schedule overview, schedule management and action templates

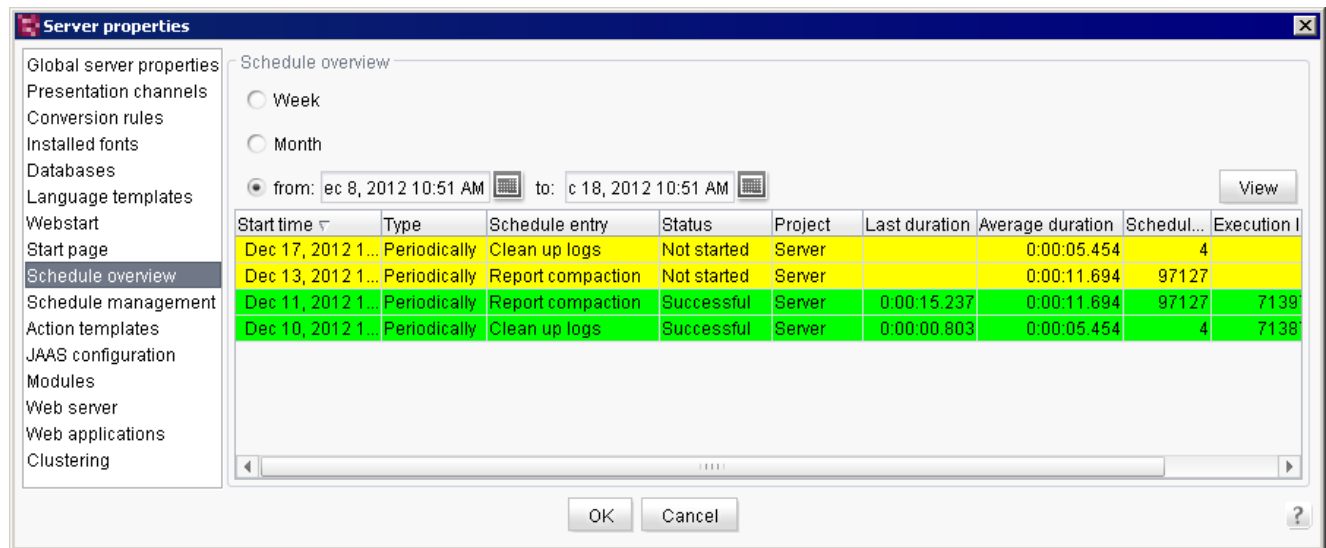


Figure 7-46: Server properties – Schedule overview

For information on the schedule overview, see Chapter 7.5.1, starting on page 372.

For information on schedule management, see Chapter 7.5.2, starting on page 375.

For information on action templates, see Chapter 7.5.3, starting on page 378.



7.3.10 JAAS configuration

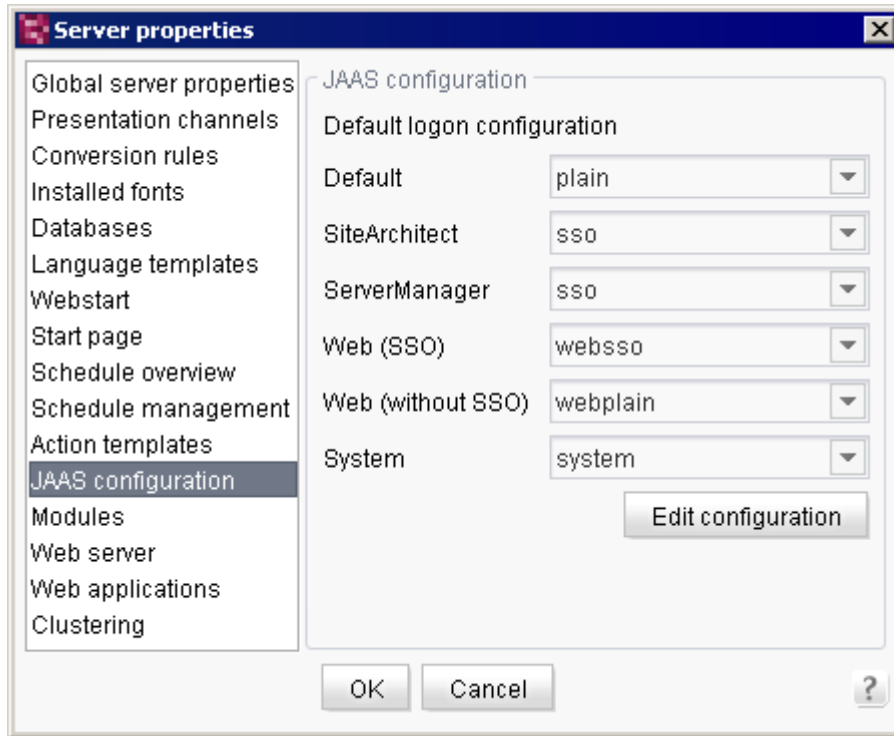


Figure 7-47: Server properties – JAAS³⁰ configuration

Important login configuration settings are defined and adjusted in this area. The settings defined here are saved in the `fs-jaas.conf` configuration file (see Chapter 4.3.4 page 85). When modifying a file in ServerManager, the file is rewritten and automatically reloaded.

The drop-down list can be used to define the respective login procedure for different login options (e.g. client login or preview request). This mapping is saved in the `fs-server.conf` configuration file covered in "Area: JAAS" (see Chapter 4.3.1.6 page 45).

Edit configuration: clicking on this button opens an editor for the `fs-jaas.conf` configuration file. The file contains a configuration example for all available login modules. The editor is used to define additional modules. Once the input is confirmed by clicking on "OK", all changes to the `fs-jaas.conf` file are saved and automatically loaded.

³⁰ Java Authentication and Authorization Service



7.3.11 Modules

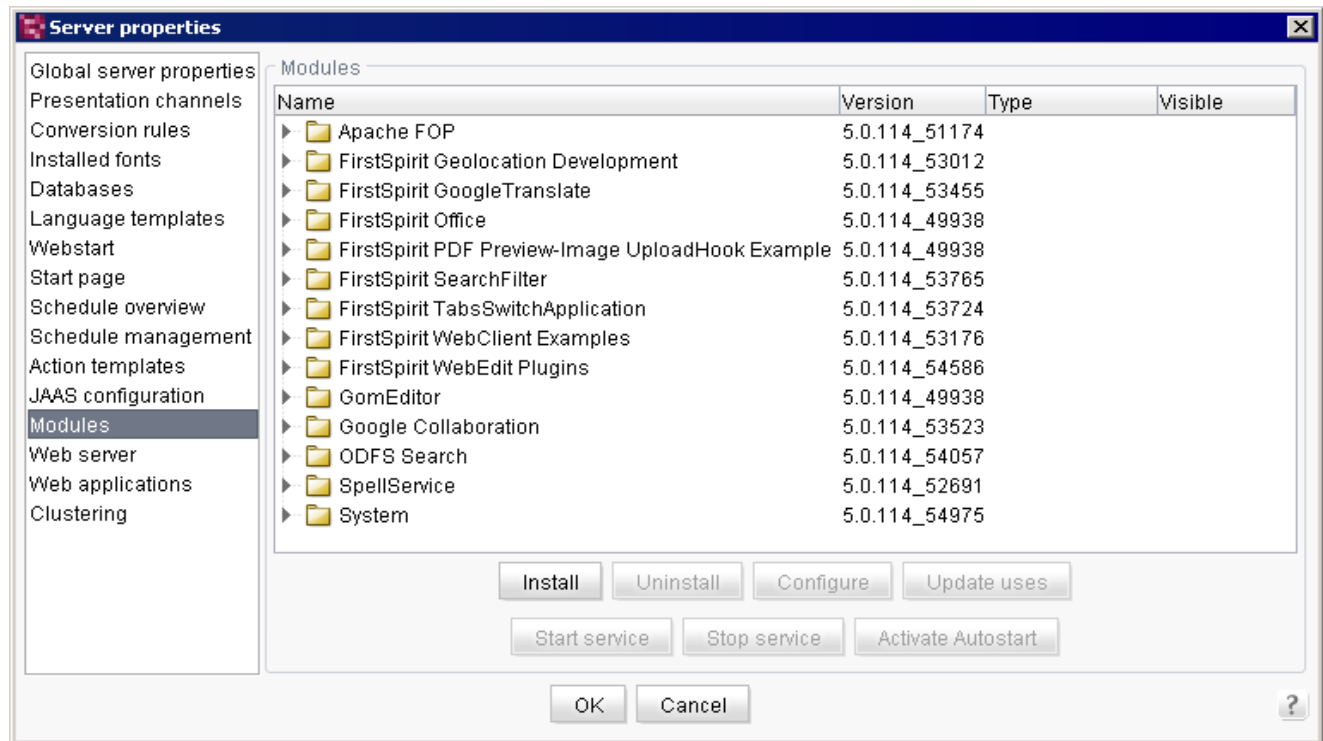


Figure 7-48: Server properties – Modules

FirstSpirit modules can be installed and configured in this area. Initially, the system module is present with the standard components. The modules are displayed in alphabetical order (by module name).



When installing and updating modules that are either the basis for data themselves or are the basis for the data via directly or indirectly dependent services, these data are no longer available to these processes until the processes are accessible after restarting (generations, clients, etc.).

Name: name of the module or name of a component of the module. The modules can be identified in the dialog by the folder icon. Each module has one or more components that are displayed under the module. Components can be identified by the file icon. There are different types of components (see "Type").

Version: the version number of the module or of the component installed on the FirstSpirit server.



Type: type of component. A distinction is made between the following components:

- **Library:** a library is a non-configurable collection of classes packed in one or more jar files. After installation they are available on the FirstSpirit server, within the clients, in scripts and other modules.
- **Editor:** an editor is a combination of GUI and rendering components. The editor is used to expand FirstSpirit Client to include custom input options. (Example: the CMS_INPUT_PERMISSION input component for defining permissions. This input component works with the relevant service that loads and provides the server group definitions.)
- **Service:** a service is a server component that can be activated via a public interface composed of input components or scripts. (Examples include spell checking or the CMS_INPUT_PERMISSION permissions input component service.)
- **Web application:** a web application defines JSP tags and servlets that can be used and called in projects. (DynamicPersonalization and Search are examples of web applications.)
- **Web server:** a web server component provides functions for installing and uninstalling web applications (Examples include the internal web server control or Tomcat support) (for more information, see Chapter 7.3.12 page 271).

Visible: components are available after installation only in a specific area. A distinction is made between the following areas:

- **Global:** global (system-wide) components are available on the server after installation; this ensures that these components are also available in all scripts and other components within FirstSpirit applications (Examples: all services; e.g. the permission service).
- **Project:** after installation, local project components can be added to the desired projects via their project properties (see Chapter 7.4.18 page 343). The user then has the option of configuring these components. Configurability depends on the installed component.
- **Web:** after installation, local web components can be added to the individual web areas (preview, staging, live) within the desired projects (see Chapter 7.4.18 page 343). Configurability depends on the installed component. The components can be configured differently for the respective projects.



Install/Update: clicking on the "Install" button opens a file selection dialog. Here you can select the fsm file that is to be installed (such as the FirstSpirit DynamicPersonalization module). The successfully installed file is then displayed in the "Server properties" dialog (see Figure 7-48).

If the installed component contains a service, a dialog where the Autostart option for the service can be configured is also displayed. If the dialog is confirmed by clicking on "Yes", the service is started automatically every time the server is restarted (see Activate/deactivate Autostart):

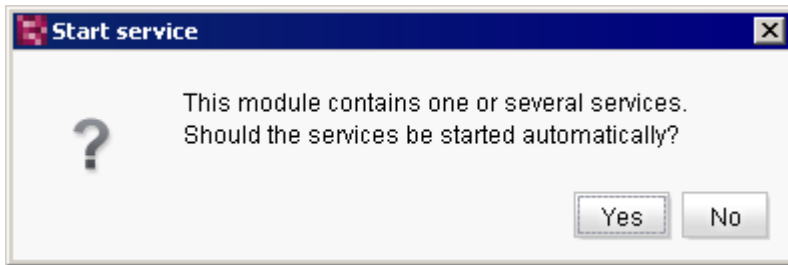


Figure 7-49: Installing Autostart for services

If a newer version of the module is available, the module can also be updated using the "Installation" button on FirstSpirit Server. It is not necessary to uninstall the module beforehand. Although the original configuration is preserved, adjustments in the configuration settings or within the projects using the module may be required. A new version of the module is therefore not imported automatically to the projects, and updating within a project must be carried out manually (see Chapter 7.4.17 page 341, and Chapter 7.4.18 page 343). The uses can be modified using the "Update uses" button.



After updating modules that have dependencies to modules with services, these services need to be restarted manually. (For information on using the "Stop services" / "Start services" buttons or ServerMonitoring, see Chapter 8.6.2.4 page 482.)

Uninstall: to uninstall a particular module, select it from the overview of installed modules and click on this button. The system module cannot be uninstalled.

Only modules that are not being used in a project can be uninstalled. When attempting to uninstall a module that is still in use, an error message will appear containing a list of all projects currently using the module. The module must first be removed from these projects before it can be uninstalled (see Chapter 7.4.18 page 343).

Configure: use this button to configure the selected component. Configurability depends on the selected component (see "Spellservice" for an example). Permissions can be set for a module as well (see Chapter 7.3.11.1 page 269):



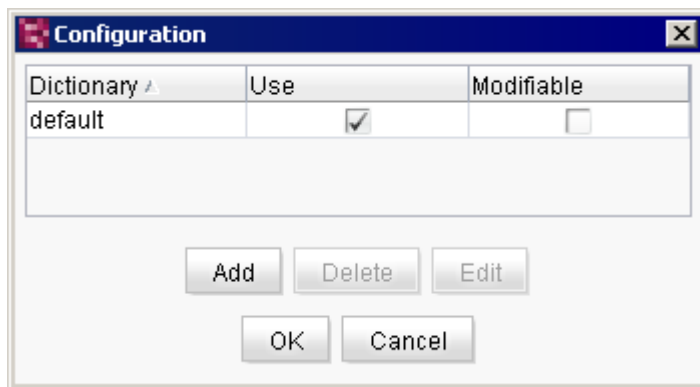


Figure 7-50: Spellservice configuration example

Update uses: the "Update uses" button is available for convenient updating of a module contained a project or web application. Clicking on the button opens a dialog box where the project or web applications for all projects that have been using this module can be updated:

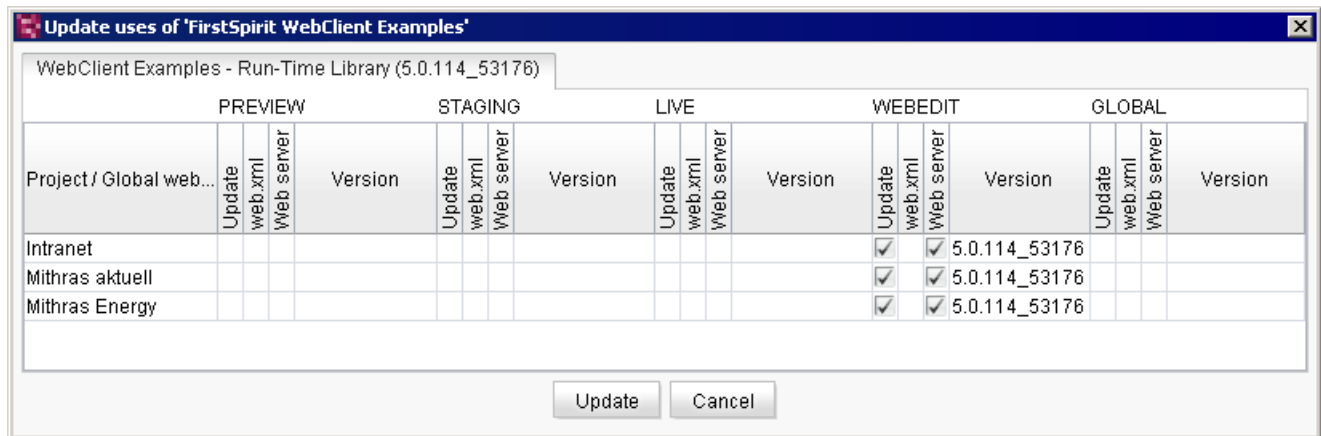


Figure 7-51: Updating all uses of a module

Updating projects does not have to be carried out only through their individual project properties; the update can also be controlled centrally using the server properties.

Start service: click on this button to start a service. A service can also be started in ServerMonitoring (see Chapter 8.6.2.4 page 482).



After updating modules that have dependencies to modules with services, these services need to be restarted manually.

Stop service: click on this button to stop a started service. A service can also be stopped in ServerMonitoring (see Chapter 8.6.2.4 page 482).



Activate/deactivate Autostart: click on this button to activate or deactivate automatic starting of a service. If the Autostart option is activated, the service is automatically started every time the server is restarted. If the Autostart option is deactivated, the service must be started manually every time the server is restarted. This setting can also be set initially during the installation or updating of services.



Many of the above mentioned functions are available in the FirstSpirit Developer API too (interface `ModuleAdminAgent`, package `de.espirit.firstspirit.agency`).

7.3.11.1 Trusted modules

FirstSpirit SiteArchitect and First Spirit ServerManager run using a JNLP file, i.e. over Java Web Start. This results in limitations when using some functions for modules not signed by e-Spirit or classes in the jar archives. Java programs usually run in a "sandbox". This means that they do not have complete access to the computer (and its resources) on which they are run. Access to local resources such as files, the clipboard, network, etc. is through a security manager.

The internal FirstSpirit modules are signed with an "e-Spirit AG" key. This is a component of the internal FirstSpirit security policy. In addition, the key is confirmed by a root authority that is in turn recognized by the Java certificate manager.

External components or modules that access security-related functions can be configured easily using FirstSpirit ServerManager. Each module installed (except for the FirstSpirit system modules) can optionally include permissions to the local system resources. This makes it possible to trust a module that carries out security-related operations, i.e. access to the clipboard (`java.awt.AWTPermission ClipboardAccess`). Permissions to carry out operations can be assigned to this module. This takes place internally through the FirstSpirit Security Manager/ClassLoader.

The configuration screen where the module permissions are set in the FirstSpirit ServerManager appears as follows:



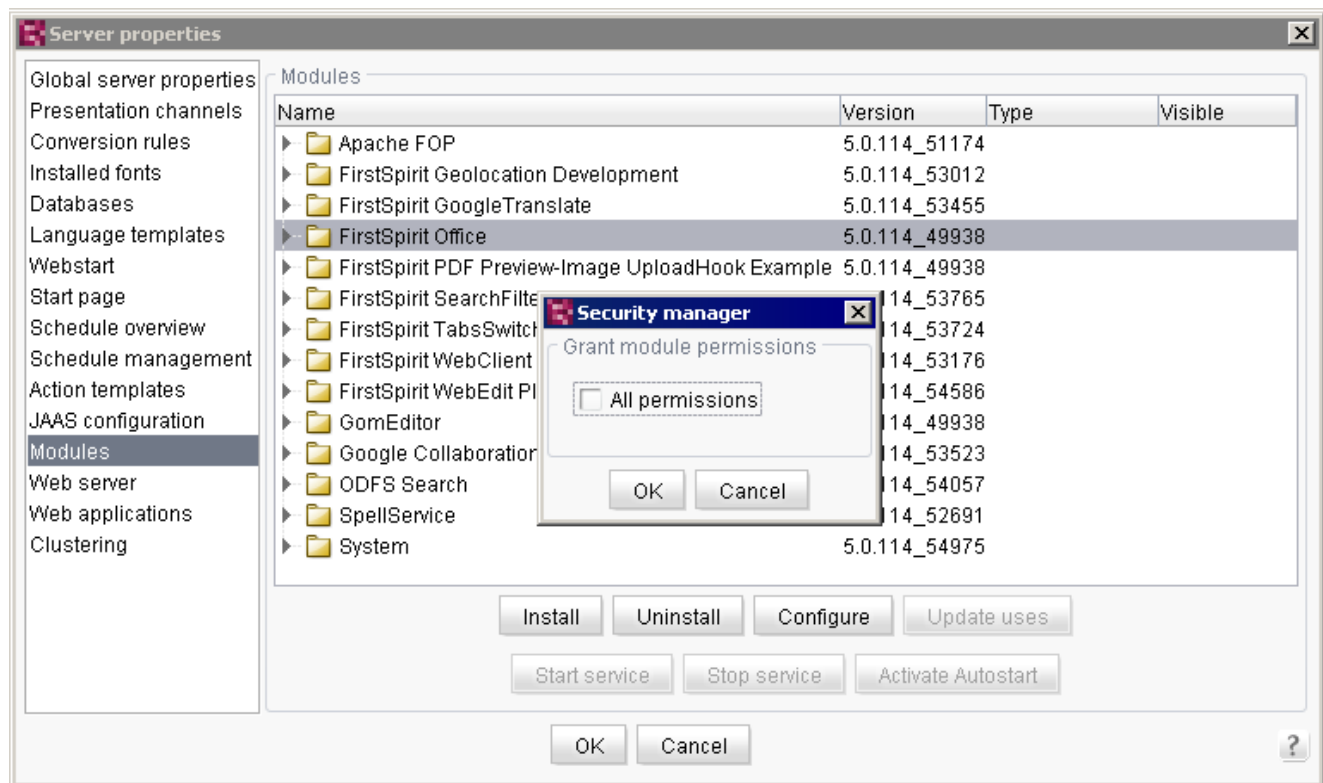


Figure 7-52: Security manager – Granting all permissions to this module



If an external component or a module is classified as trusted, there is no assurance that the FirstSpirit secure access mechanisms will be fully effective. Any malfunctions that may occur can no longer be attributed to a clear cause, making an error difficult or impossible to diagnose. As part of FirstSpirit product maintenance, system configurations classified as trusted external components or modules are therefore not permitted.

For more information, see the "FirstSpirit Manual for Developers (Components)" (German only).



7.3.12 Web server

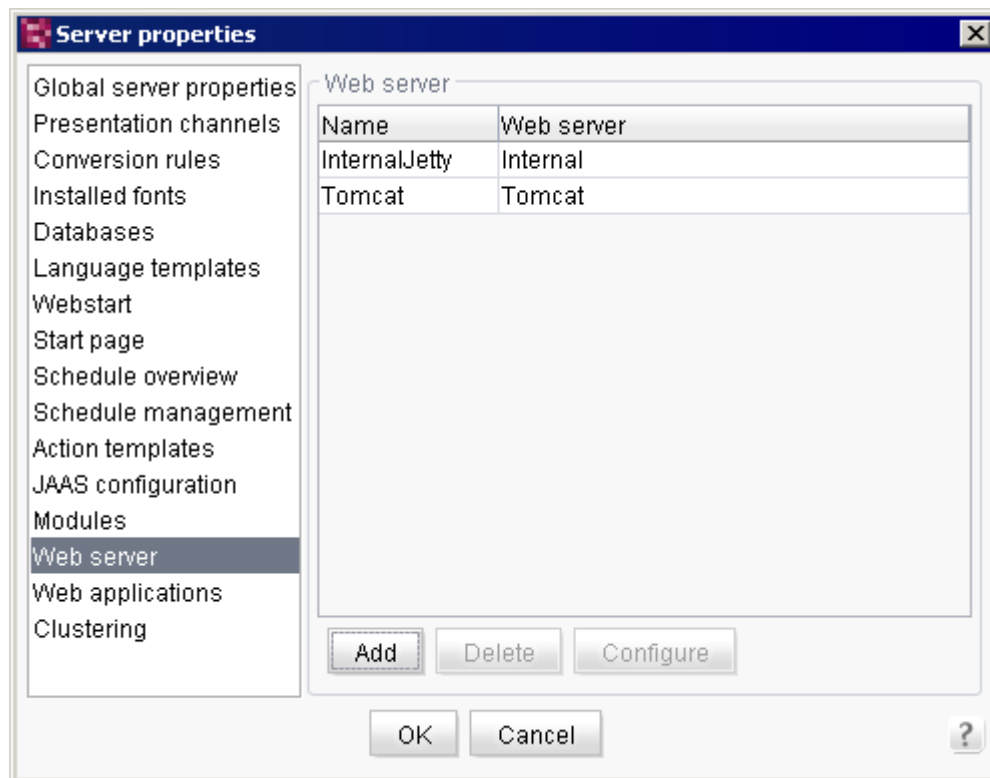


Figure 7-53: Server properties – Web server

Web server control can be added here for installation and removal procedures within the project areas (see Chapter 7.4.17 page 341).

Control for the internal Jetty web server is provided by default. The relevant entry is displayed in the table. This web server control cannot be changed or deleted and is available automatically immediately after FirstSpirit is installed.

In addition to this default entry, any generic or external web server controls, including Tomcat web server, can be added and configured.

Name: a unique name under which the web server instance was added.

Web server: the "System" default module contains four different types of web servers:

- Internal (internal web server): in the case of the internal Jetty web server, control (installation or removal of web applications) is available immediately after installation of FirstSpirit.
- External (external web server): these are external web servers that are not supported by FirstSpirit. Installation or removal of web applications on the web server as well as all other configuration settings must therefore be done manually.



- Tomcat: this is an external web server that is supported by FirstSpirit. The web applications can be installed and updated automatically on the web server using the FirstSpirit interface.
- Generic (local web server): FirstSpirit allows for the easy connection of web servers as long as it is possible to implement control (installation or removal of web applications) for these web servers using the BeanShell script (see Chapter 7.3.12.2 page 273).

Clicking the "Add" button adds a new generic or external web server instance to FirstSpirit Server.

- Internal web server: "InternalJetty" is available by default and does not need to be added.
- Generic web server (see Chapter 7.3.12.1 page 273)
- External web server (see Chapter 7.3.12.3 page 275)
- Tomcat (see Chapter 7.3.12.5 page 276)

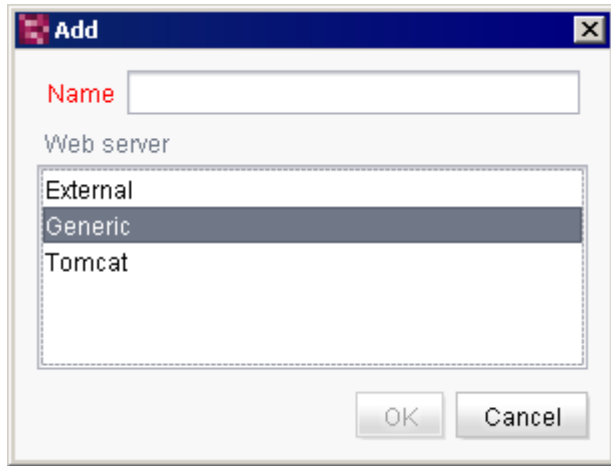
Clicking the "Delete" button deletes an added web server instance. The default "InternalJetty" instance cannot be deleted.

Clicking the "Configure" button opens the web server configuration dialog.

- Internal web server: this configuration cannot be edited.
- Generic web server: if it is a generic web server, the necessary functions for installation and removal can be configured here (see Chapter 7.3.12.2 page 273).
- External web server: configuration for external web servers is not supported and must be done manually (see Chapter 7.3.12.4 page 276)
- Tomcat: configuration for Tomcat web servers is supported and can be done automatically (see Chapter 7.3.12.6 page 277).



7.3.12.1 Adding a generic web server

**Figure 7-54: Adding a generic web server**

Name: a unique name for the particular web server instance must be specified here.

Web server: adding a new generic web server instance is possible by selecting the "Generic" entry.



After adding a new generic web server instance, the installation and removal functionality needs to be implemented (see Chapter 7.3.12.2 page 273).

7.3.12.2 Configuring a generic web server

To bind a new web server to FirstSpirit Server, the following functionality is required:

1. Installation
2. Installation status verification
3. Deinstallation

The "Configure" button opens the web server configuration dialog.



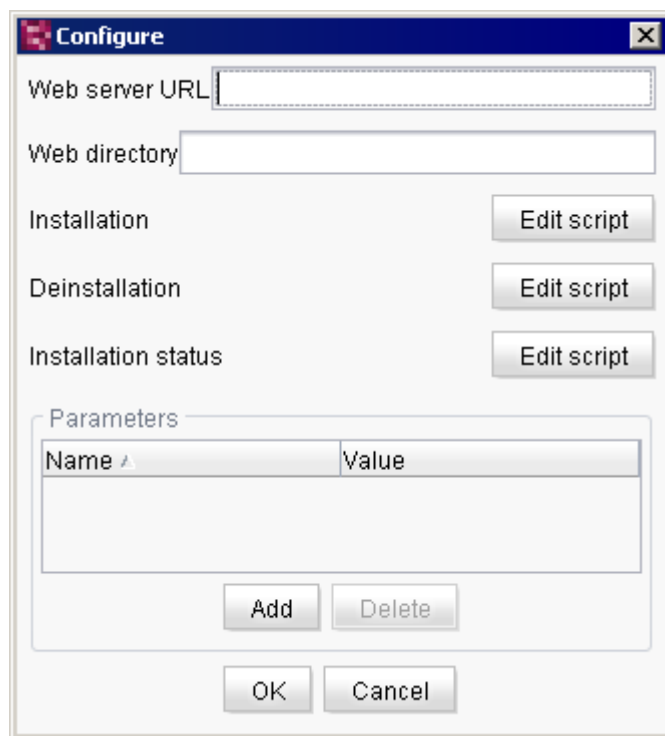


Figure 7-55: Configuring web server functions

Web server URL: the URL of the new web server can be entered in this field. The URL is required, for instance, to adapt links for FirstSpirit applications on the start page.

Web directory: the path to the web directory of the generic web server for the use of FirstSpirit applications (e.g. fs5staging). If the generic web server is used for the fs5staging web application, for instance, the project files are generated in the specified directory of the generic web server. The path is also required in order to run the configured scripts (e.g. installation or removal).

Edit script: all functions can be enabled using a BeanShell script. The "Edit script" button (next to the particular function) opens the script dialog.

The desired functions can be carried out under the control of scripts for each web server supported by FirstSpirit. The required parameters for the scripts can be added in the Parameters area. The functionality should be checked using the "Test" button before saving. After saving the script, the functionality will be available in the "Web applications" or "Web components" area:

- Server properties / Web applications (see Chapter 7.3.13 page 278)
- Project properties / Web components (see Chapter 7.4.18 page 343).



The web directory containing the generic web server can be specified (see "Web directory") for running the scripts.

Add parameter: in the "Parameters" area (see Figure 7-55), parameters can be added, which are then available in all script contexts (from installation to removal; example: path information).

Delete parameter: clicking on this button removes a highlighted parameter from the list.



Deleting parameters can result in the inability to work with or adapt the scripts.

7.3.12.3 Adding an external web server

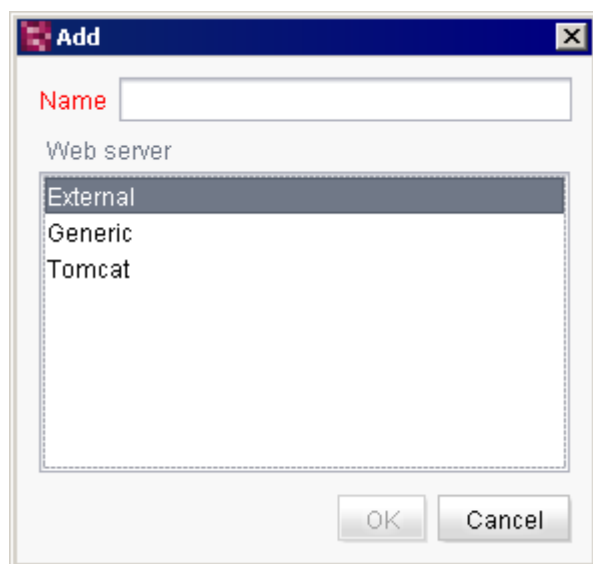


Figure 7-56: Adding an external web server

Name: a unique name for the particular web server instance must be specified here.

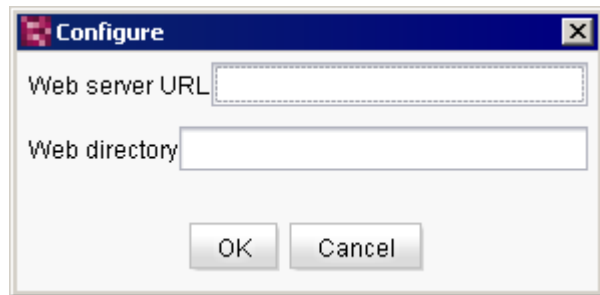
Web server: adding a new external web server instance is possible by selecting the "External" entry.



After adding a new external web server instance, the URL of the web server can be specified (see Chapter 7.3.12.4 page 276). Additional support (installation or removal) is not provided and must be done manually.



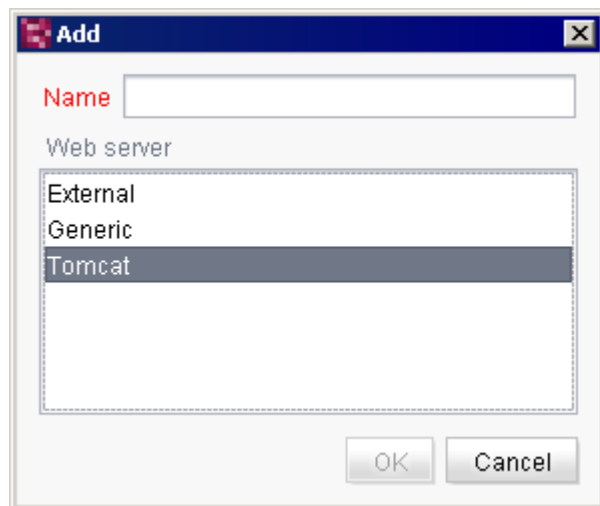
7.3.12.4 Configuring an external web server

**Figure 7-57: Configuring an external web server**

Web server URL: the URL of the new web server can be entered in this field. The URL is required, for instance, to adapt links for FirstSpirit applications on the start page.

Web directory: the path to the directory of the web server can be entered in this field. This is required, for instance, for deleting staging and preview directories when the related project is deleted. The path is also read out and used when updating web applications.

7.3.12.5 Adding a Tomcat web server

**Figure 7-58: Adding a Tomcat web server**

Name: a unique name for the particular web server instance must be specified here.

Web server: adding a new Tomcat web server instance is possible by selecting the "Tomcat" entry.





After adding a new Tomcat web server instance, the installation and removal functionality must be configured (see Chapter 7.3.12.6 page 277).



For more information on how to use Tomcat, see Chapter 4.5 page 113).

7.3.12.6 Configuring a Tomcat web server

The "Configure" button opens the web server configuration dialog.

The screenshot shows a standard Windows-style dialog box titled "Configure". It contains five text input fields arranged vertically, each with a label to its left: "Web server URL", "Web directory", "Tomcat user", "Tomcat password", and "Tomcat manager URLs". At the bottom of the dialog, there are two buttons: "OK" and "Cancel". The dialog box has a standard title bar with a close button (X) in the top right corner.

Figure 7-59: Configuring a Tomcat web server

Web server URL: the URL that is preconfigured when called must be entered in this field URL, e.g. `http://tomcat:123/fs5webedit`. This URL is required, for instance, to adapt links for FirstSpirit applications on the start page.

Web directory: the path to the web directory of the Tomcat web server must be entered in this field in order to use FirstSpirit applications (e.g. `fs5staging`). If the Tomcat web server is used for the `fs5staging` application, for instance, the project files are generated in the specified directory of the Tomcat web server. If it is not present, the web directory entered here will be added during generation.

Tomcat user: the user name that can be used to access to the Tomcat Manager URL (see below) must be entered in this field. The user name and password are specified during installation of the Tomcat server and can be found in the `tomcat-users.xml` file in the `conf`



Tomcat web server directory.

Tomcat password: the password for the user name previously specified must be entered in this field. The password provides access to the Tomcat Manager URL (see below). The user name and password are specified during installation of the Tomcat server and can be found in the `tomcat-users.xml` file in the "conf" Tomcat web server directory.

Tomcat manager URLs: Tomcat Manager can be used to view and manage web applications. In addition to an HTML interface, it also provides a text interface. The URL for the Tomcat Manager text interface must be entered in this field; for instance,

`http://localhost:8080/manager`

for Tomcat 6

or

`http://localhost:8080/manager/text`

for Tomcat 7

Multiple URLs can be entered and separated by commas.

7.3.13 Web applications

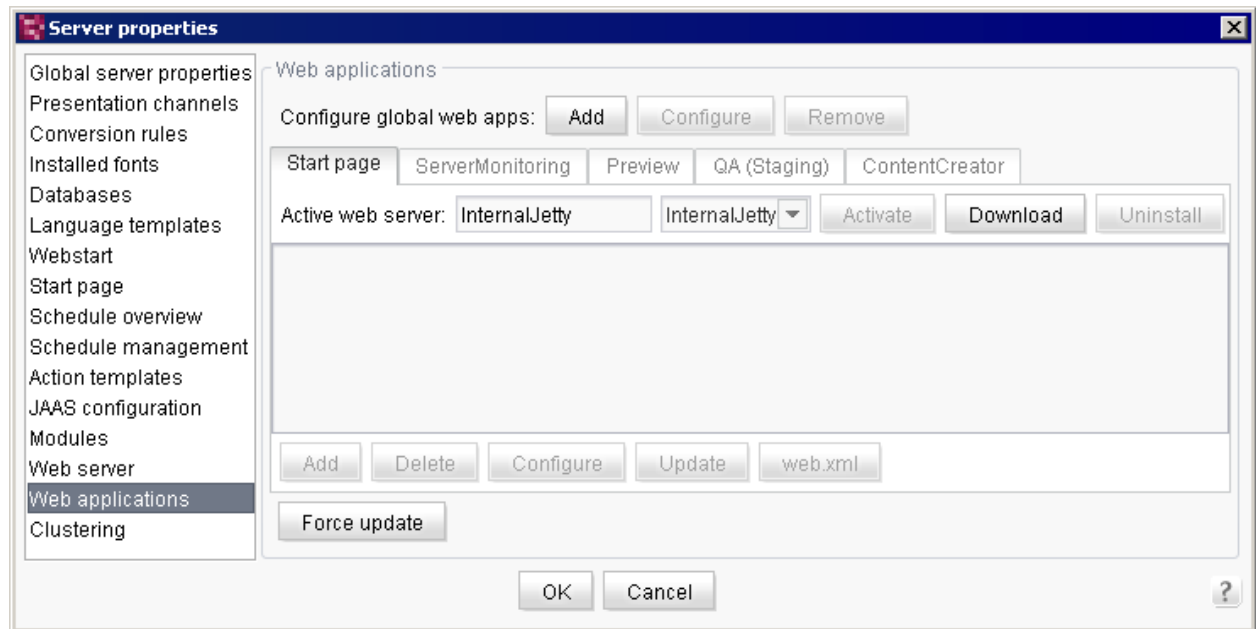


Figure 7-60: Server properties – Web applications



FirstSpirit web applications can be configured in this area. Some web applications can also be defined and configured, which are then available in all projects on the server (for information on "global" configuration, see Chapter 7.3.13.1 page 279). For instance, custom web applications for the FirstSpirit AppCenter (see the *FirstSpirit AppCenter* documentation) can be installed.

7.3.13.1 Installing global web applications

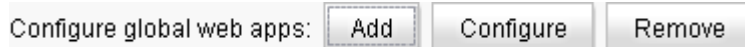


Figure 7-61: Configuring global web apps

The "Add" button is used to install a new web application on the server.

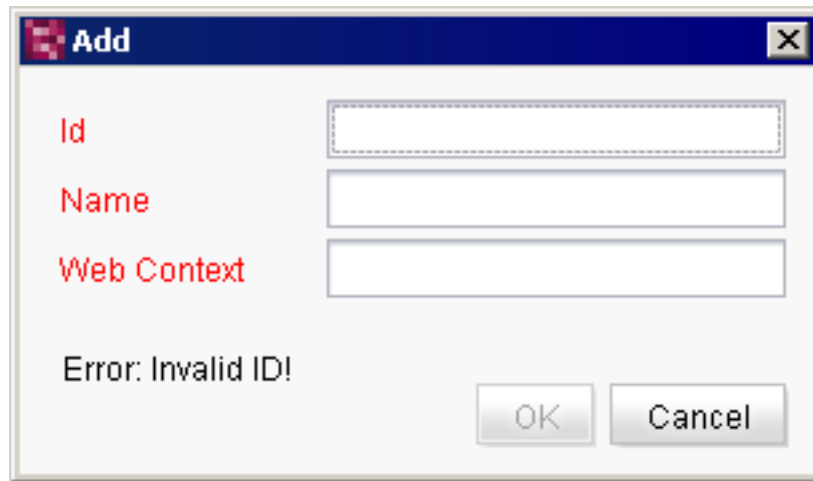


Figure 7-62: Adding a new web application

Id: a unique identifier must be entered here for the web application. This identifier is used to create a subdirectory on the server for web applications. Only lower/uppercase letters, numbers and the underscore character may be used. The ID can never be changed once it is saved.

Using a web application ID via the FirstSpirit Developer API, the URL to a global web application can be determined using the `de.espirit.firstspirit.agency.LegacyModuleAgent` interface.

Name: the name used as the display name must be entered here. This can be changed at a later time, if necessary.

Web Context: a context name, which forms part of the web application URL, must be entered here. The name must not match the name of any FirstSpirit web applications already included by default (e.g., *fs5root*, *fs5preview*, *fs5staging*, *fs5webedit*, *fs5webmon*).



7.3.13.2 Configuring web applications

Every web application is displayed and configured in its own tab. The default web applications are:

Start page / Preview: configuration for the FirstSpirit start page and the preview (see Chapter 6 page 194).

QA (Staging): configuration for the generated project QA (Staging) contents.

ContentCreator: configuration for the ContentCreator editing environment.

ServerMonitoring: configuration for FirstSpirit ServerMonitoring (see Chapter 8 page 449)

- A web server can be set for each web application:



Any web servers that were configured in the "Web server" area can be selected (see Chapter 7.3.12 page 271).

Different conversion steps are required depending on the type of web server:

- Internal web server (see Chapter 7.3.13.3 page 281)
- Generic web server (see Chapter 7.3.13.4 page 281)
- External web server (see Chapter 7.3.13.5 page 282)

Activate: the "Activate" button switches the configuration of the web application to the selected web server, which is then displayed as the active web server for the application. The links to the web application (e.g. on the start page) are automatically switched to the new web server.

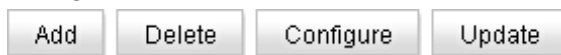
Install: the "Install" button installs the web application on the selected web server. If the button is disabled, the web application is already installed. If the web server selected is an external web server or a generic web server (without the required script functionality), the "Download" button will be displayed instead (see Chapter 7.3.13.5 page 282).

Download: the "Download" button is used to download an application's WAR file, which needs to be installed manually on the web server. The button is only displayed in order to configure external web servers or generic web servers (without the required script functionality).

Uninstall: the "Uninstall" button removes the web application from the selected web server. If the button is disabled, the web application has not been installed yet. If a web application is uninstalled, the relevant entry is removed from the fs-server.conf configuration file.



- For each web application, web components on the server can be added, removed, configured and updated:



- The file `web.xml` can be edited manually for each web application:

A small rectangular button with rounded corners, labeled 'web.xml'.

The web components for a web application can be pooled together and installed on the web server or downloaded as a WAR file.

The function is similar to that of the web components in the project properties. Also refer to Chapter 7.4.18 page 343 in this regard.

7.3.13.3 Configuring an internal web server for a web application

Control for the internal Jetty web server is provided by default and cannot be changed. The internal web server is activated for all FirstSpirit web applications after FirstSpirit is installed.

If a different web server was activated for a FirstSpirit web application, the configuration can be reset to the internal web server by taking the following steps:

1. Select "internalJetty" from the combo box.
2. The "Install" button becomes active. Clicking on this button installs the web application on the internal web server.
3. The "Activate" button will be active after installation. Clicking on this button switches the configuration of the web application to the selected web server, which is then displayed as the active web server for the application. The links to the web application (e.g. on the start page) are automatically switched to the new web server.
4. All changes to the configuration must be confirmed and saved by clicking on "OK".

7.3.13.4 Configuring a generic web server for a web application

It is only possible to select a generic server if a corresponding web server instance was previously added to the server (see Chapter 7.3.12.1 page 273). Control of the generic web server is not provided by default, but it is possible using scripts (see Chapter 7.3.12.2 page 273). If these scripts are not available, the procedure is identical to that of the external web server (see Chapter 4).



If a generic web server is to be activated for a FirstSpirit web application (e.g., Tomcat), the following steps are required:

1. Select the entry from the combo box for the desired generic web server.
2. The "Install" button becomes active. As long as the relevant functionality has been provided by a script, the web application can be installed on the generic web server by clicking on this button.
3. The "Activate" button will be active after installation. Clicking on this button switches the configuration of the web application to the generic web server, which is then displayed as the active web server for the application. The links to the web application (e.g. on the start page) are automatically switched to the new web server. The URL entered under "Web server URL" is used (see Figure 7-55).
4. All changes to the configuration must be confirmed and saved by clicking on "OK".

7.3.13.5 Configuring an external web server for a web application

It is only possible to select an external server if a corresponding web server instance was previously added to the server (see Chapter 7.3.12.3 page 275). Control of an external web server is not supported by FirstSpirit and must be done manually (see Chapter 7.3.12.4 page 276).

If an external web server is to be activated for a FirstSpirit web application, the following steps are required:

1. Select the entry from the combo box for the desired external web server.
2. The "Download" button becomes active. Use this button to download the application's WAR file.
3. The WAR file must be installed manually on the external web server. The installation is either done manually via the administrative interface of the external web server or automatically from the web server file system. The start page URL should be on its own virtual server so that the FirstSpirit start page in the `/fs5root` web application can be defined as the root application and so that it can be accessed, for instance, directly from `http://fs5.yourdomain.net`. However, a URL with a prefix such as `http://appserver.yourdomain.net/fs5` is also possible.
The WAR file must be reinstalled every time FirstSpirit Server is updated.
4. After installation, the web application configuration can be switched to the external web server by clicking on the "Activate" button. The external web server is now displayed as the active web server for the application. The links to the web application (e.g. on the start page) are automatically switched to the new web server. The URL entered under "Web server URL" is used (see Figure 7-57).



5. All changes to the configuration must be confirmed and saved by clicking on "OK".

7.3.13.6 Configuring a Tomcat web server for a web application

It is only possible to select a Tomcat server if a corresponding web server instance was previously added to the server (see Chapter 7.3.12.5 page 276). Control of a Tomcat web server is supported by FirstSpirit and takes place automatically (see Chapter 7.3.12.6 page 277). Installation on the web server and registering the web components is handled by FirstSpirit.

If a Tomcat web server is to be activated for a project-specific web area, the following steps are required:

1. Select the entry from the combo box for the desired Tomcat web server.
2. The "Install" button becomes active. This button copies the WAR file, which is automatically unpacked by the web server before the web components are registered.
3. The "Activate" button will be active after installation. Clicking on this button switches the configuration of the web area to the Tomcat web server, which is then displayed as the active web server for this area.
4. All changes to the configuration must be confirmed and saved by clicking on "OK".



If the `fs-server.jar` file has been updated, the steps described previously must be repeated manually for all web applications in all projects for which a "Tomcat" type web server was selected. Updates are not automatic.



7.3.14 Clustering

In this case, clustering means load balancing for generating projects on additional FirstSpirit servers ("horizontal scalability").

For information on the concept and a description of the architecture, see Chapter 7.6 page 422.

The configured FirstSpirit servers within the cluster are listed under the server properties in the "Clustering" area of FirstSpirit ServerManager. This area is also used under Windows for the initial installation of the cluster. Under Unix, the cluster is installed and configured only via the FirstSpirit Server configuration files.

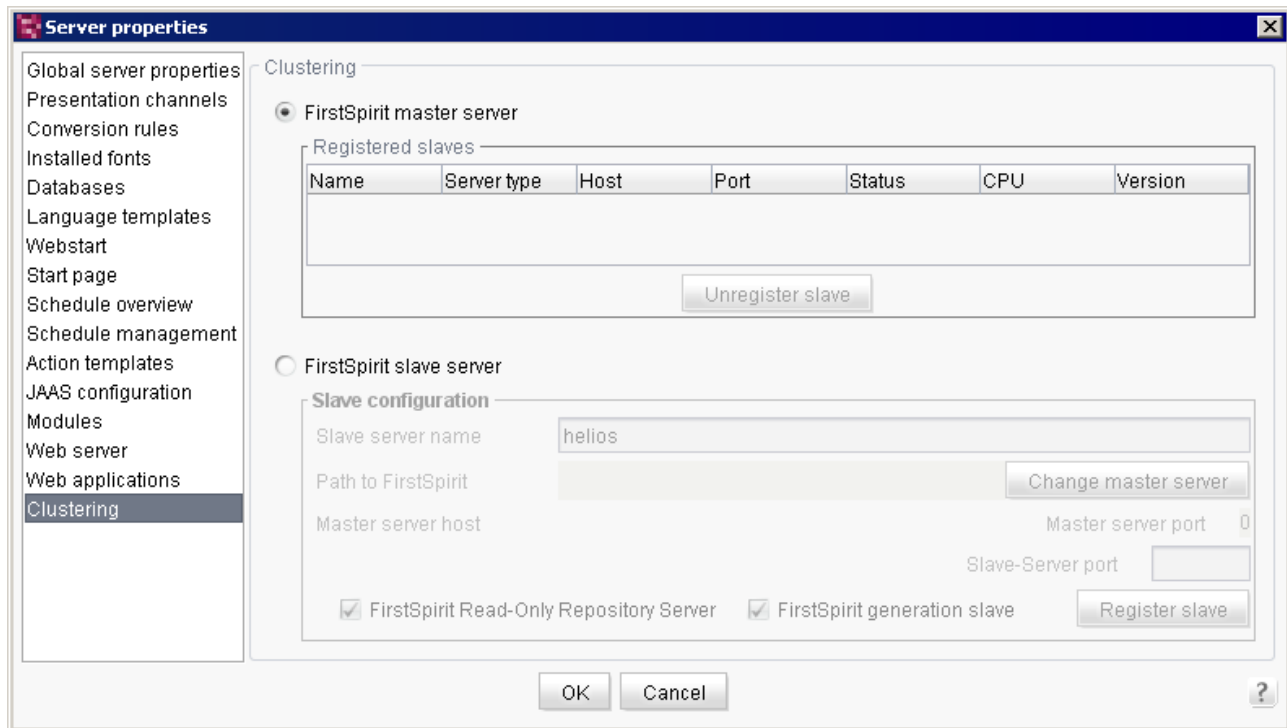


Figure 7-63: Server properties – Clustering

- FirstSpirit master server configuration under Unix: Chapter 7.3.14.1
- FirstSpirit slave server configuration under Unix: Chapter 7.3.14.2
- FirstSpirit master server configuration under Windows: Chapter 7.3.14.3
- FirstSpirit slave server configuration under Windows: Chapter 7.3.14.4



7.3.14.1 FirstSpirit master/slave server configuration under Unix

A FirstSpirit server needs to be installed first, as described in the Installation Instructions. The installation path used in this example will be `/opt/firstspirit5` and the host name will be `fs5host`. After installation, the FirstSpirit server needs to be shut down so that it can be defined as a master server by modifying the configuration as described below and so that data can be copied to the FirstSpirit slave servers.

The master server contains all of the functions of a standard FirstSpirit server and should be set up using fail-safe hardware, since all FirstSpirit clients use the master server as the endpoint in network connections. The slave servers are not self-contained and cannot be used on their own. However, all FirstSpirit functions are provided by the master server without any need for changes to the configuration should the slave servers fail. The generation schedule entries are then automatically executed on the master instead of the slave, which only results in a higher load for the master.

The following line must be added to the file `/opt/firstspirit5/conf/fs-server.conf`:

```
HOST=fs5host
```

A load-balanced file system is required in order to copy data to the slave servers and to write from the slave servers to the master server log files. NFS is available under Unix for this purpose. The NFS server should run on the same system as the FirstSpirit master server so that the master server can write to the local file system. The example provided uses Solaris. The NFS configuration can be used on other Unix systems if desired. The distinction made between read-only and read/write access directories serves to increase the operating security and data throughput by distributing only the log files across the NFS for write access. The central repository files on the master server are distributed for read access only.

The following lines need to be added to the file `/etc/dfs/dfstab` so that some selected master server directories can be released over the NFS:



```
share -F nfs -o ro /opt/firstspirit5/bin
share -F nfs -o ro /opt/firstspirit5/conf
share -F nfs -o ro /opt/firstspirit5/server
share -F nfs -o ro /opt/firstspirit5/shared
share -F nfs -o ro /opt/firstspirit5/data/projects

share -F nfs -o rw /opt/firstspirit5/data/schedule
share -F nfs -o rw /opt/firstspirit5/web/fs5staging
share -F nfs -o rw /opt/firstspirit5/log

share -F nfs -o rw /opt/firstspirit5/archive

share -F nfs -o rw /opt/firstspirit5/backup

share -F nfs -o rw /opt/firstspirit5/export
```

If the security requirements are higher, release over the NFS should be limited to exclusive access for FirstSpirit slaves by using the authentication procedure (IP address, Kerberos, etc.) supported by the operating system.

The NSF server must now be notified of the changed configuration, i.e. by using `shareall`.

The FirstSpirit master server can then be started via `fs5 start` (or `svcadm enable fs5`). The slave server availability and load can be monitored on the FirstSpirit start page by way of the "Clustering" area of FirstSpirit ServerMonitoring (see Chapter 8.6.6 page 493).

The following master server log files are used for error analyses when problems arise during and after configuration:

```
/opt/firstspirit5/log/fs-wrapper.log
/opt/firstspirit5/log/fs-server.log
```

7.3.14.2 FirstSpirit slave server configuration under Unix

A FirstSpirit server needs to be installed first, as described in the Installation Instructions. Please deactivate the automatic import of the example project by specifying `FSDEMO=false` (see *FirstSpirit Installation Instructions*). When doing so, it is important to note that the Unix user ID used for the FirstSpirit installation is the same one used on the master server. Otherwise, release and write access of shared NFS directories become more complicated. The installation path used in this example will be `/slave/firstspirit5` and the host name will be `fs5slave`. After installation, FirstSpirit Server needs to be shut down so that it can be defined as a slave server by modifying the configuration as described below and so that data can be copied to the FirstSpirit master servers.



To simplify subsequent updating to a new version of FirstSpirit, some JAR files and start scripts are used by the master server directly over NFS. After the master servers are updated, the slave servers therefore only need to be restarted so that files do not have to be replaced manually.

For configuration, changes are then made to the `wrapper.mainclass` and `#include` parameters already present in the file `/slave/firstspirit5/conf/fs-wrapper.conf`:

```
wrapper.java.mainclass=de.espirit.firstspirit.server.ClusterHost  
  
#include ../conf/fs-wrapper-license.slave.conf
```

and the following parameters are added:

```
wrapper.java.additional.12=-Dcmsroot=/import/fs5master  
  
wrapper.java.additional.13=-Dnode=Generierungs-Slave  
  
wrapper.java.additional.14=-DinitialPort=1088
```

The port number specified for `initialPort` defines the TCP port that the slave server uses to receive master server control data.

The file created automatically for the standard installation of FirstSpirit `/slave/firstspirit5/conf/fs-server.conf` is not required for the slave server and should be deleted:

```
rm /slave/firstspirit5/conf/fs-server.conf
```

The NFS client configuration shown in this example is running under Solaris. This configuration can be used on different Unix systems as desired.

First the mount points are created on which the master files are provided:

```
cd /slave/firstspirit5  
rm -r bin server shared  
mkdir bin server shared  
  
mkdir -p /import/fs5master/conf  
mkdir -p /import/fs5master/data/projects  
mkdir -p /import/fs5master/data/schedule  
mkdir -p /import/fs5master/web/fs5staging  
mkdir -p /import/fs5master/log  
  
mkdir -p /import/fs5master/archive  
  
mkdir -p /import/fs5master/backup  
  
mkdir -p /import/fs5master/export
```



The following entries are required in the `/etc/vfstab` file so that the directories can be used by the master servers:

```
fs5host:/opt/firstspirit5/bin - /slave/firstspirit5/bin nfs - yes ro,hard,intr
fs5host:/opt/firstspirit5/server - /slave/firstspirit5/server nfs - yes ro,hard,intr
fs5host:/opt/firstspirit5/shared - /slave/firstspirit5/shared nfs - yes ro,hard,intr
fs5host:/opt/firstspirit5/conf - /import/fs5master/conf nfs - yes ro,hard,intr
fs5host:/opt/firstspirit5/data/projects - /import/fs5master/data/projects nfs - yes ro,hard,intr
fs5host:/opt/firstspirit5/data/schedule - /import/fs5master/data/schedule nfs - yes rw,hard,intr
fs5host:/opt/firstspirit5/web/fs5staging - /import/fs5master/web/fs5staging nfs - yes rw,hard,intr
fs5host:/opt/firstspirit5/log - /import/fs5master/log nfs - yes rw,hard,intr
fs5host:/opt/firstspirit5/archive - /import/fs5master/archive nfs - yes rw,hard,intr
fs5host:/opt/firstspirit5/backup - /import/fs5master/backup nfs - yes rw,hard,intr
fs5host:/opt/firstspirit5/export - /import/fs5master/export nfs - yes rw,hard,intr
```

The NSF client must now be notified of the changed configuration, i.e. by using `mount -F nfs -a`.

The FirstSpirit slave server can then be started via `fs5 start` (or `svcadm enable fs5`). The slave server availability and load can be monitored on the FirstSpirit start page by way of the "Clustering" area of FirstSpirit ServerMonitoring (see Chapter 8.6.6 page 493).

The following slave server log files are used for error analyses when problems arise during and after configuration:

```
/slave/firstspirit5/log/fs-wrapper.log
/import/fs5master/log/fs5slave/fs-server.log
```



To ensure that a slave server is used during project generation, the "Execute on cluster" option must be selected under the properties of the particular schedule entry configuration (see Chapter 7.6.4 page 425).

7.3.14.3 Master server configuration under Windows

If the "FirstSpirit master server" radio button is selected (see Figure 7-63), it is a FirstSpirit master server. This server manages all FirstSpirit Server projects centrally and whenever possible distributes tasks to other FirstSpirit servers. The available servers (e.g. generation slaves) are displayed in the "Registered slaves" area (see Figure 7-64):



Name	Server type	Host	Port	Status	CPU	Version
cluster1	SLAVE	myServer	5051	IDLE	0%	FirstSpirit Ver

Unregister slave

Figure 7-64: Configuration as FirstSpirit master server

The "Unregister slave" button is used to remove a "Registered slave" from the selection list.

7.3.14.4 Slave server configuration under Windows

The FirstSpirit server can be defined as a "slave server" in the bottom section of the dialog. To do this, a FirstSpirit master server must be specified in the "Slave configuration" section. After successful configuration, the FirstSpirit server is available as a member of a cluster group.

Slave configuration

Slave server name: cluster1

Path to FirstSpirit: [Empty field] Change master server

Master server host: [Empty field] Master server port: 0

Slave-Server port: [Empty field]

☒ FirstSpirit Read-Only Repository Server ☒ FirstSpirit generation slave Register slave

Figure 7-65: Configuration as FirstSpirit slave server

Slave server name: name of the FirstSpirit slave.

Path to FirstSpirit: the "Change master server" button is used to specify the path to the FirstSpirit master server. Clicking on the button opens a dialog where the path can be entered manually:





Figure 7-66: Path to FirstSpirit master server

After clicking on "OK", the program checks to see if any additional FirstSpirit server is installed. This FirstSpirit server (master server) must not be the same server as the one currently being configured ("slave server"). If an additional FirstSpirit server has been found, the parameters "Master server host" and "Master server port" are populated automatically. The parameters of the configuration file `fs-server.conf` (of the FirstSpirit master server) are used for this.

The dialog for a correctly configured slave server appears as follows:

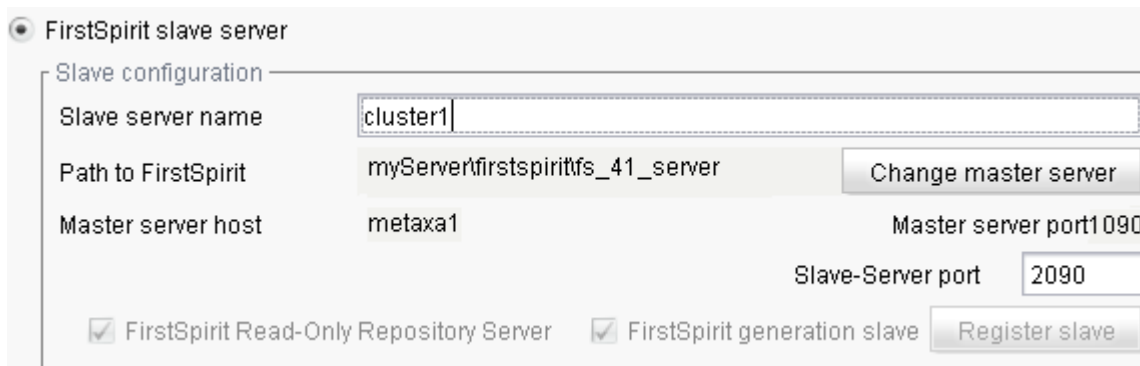


Figure 7-67: Configuration as FirstSpirit slave server

The "Change master server" button can be used to change the master server set here at a later date.

Master server host: host name of the FirstSpirit master server. The field is populated automatically after a valid path to the FirstSpirit server is entered.

Master server host: port number of the FirstSpirit master server. The field is populated automatically after a valid path to the FirstSpirit server is entered.

FirstSpirit Read-Only Repository Server / FirstSpirit generation slave: these checkboxes are selected automatically. The FirstSpirit server is added to the cluster group as a generation



server. A generation server contains a RORS³¹. The "FirstSpirit Read-Only Repository Server" checkbox is therefore also selected automatically.

Slave server port: a free port must be specified here for the slave server.

The "Register slave" button opens the following security warning:

Do you really want to register this server as "slave"? This server will be restarted in "slave" mode. The "standalone" mode can only be restored on the master server! Please note: by confirming, ServerManager will also be shut down.

Confirming the dialog by clicking on "Yes" copies the configuration settings to the corresponding configuration files (fs-server.conf, fs-wrapper.conf). The server is then shut down and restarted in "slave" mode. The new slave server's ServerManager is also closed automatically.

A slave that is already registered can only be unregistered via the server properties of the FirstSpirit master server.

³¹ ReadOnly Repository Server (see section 7.6.1, page 387)



7.4 Project properties

The project properties can be called up by double-clicking on the desired project in the project list. Additional ways of accessing the project properties are via the "Project"/"Properties" menu item on the menu bar (see Chapter 7.2.3.7 page 228) or by using the "Change properties" button.

Clicking on the "Change properties" button opens the project properties (via the project selection list) of the highlighted project.

The project selection list shows only projects for which the user who is logged in has the necessary permissions. A server administrator naturally has access to all projects on the server, and a project administrator only has access to projects for which he has been explicitly been assigned as administrator (for more information, see Chapter 7.1 page 213, and Chapter 7.4.2 page 294).

Once called, the "Edit project" dialog box appears (see Figure 7-69). The properties for the selected project can be editing using the menu entries on the left-hand side of the window. The associated settings can be changed for each menu entry on the right-hand side of the window.



The project settings for a project can always be edited only by one user at a time. An error message will appear if a second user attempts to open the project settings area for a particular project.

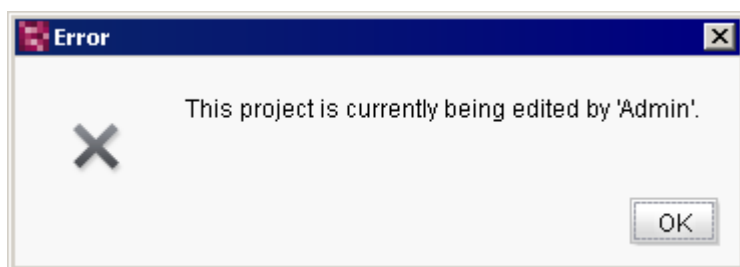
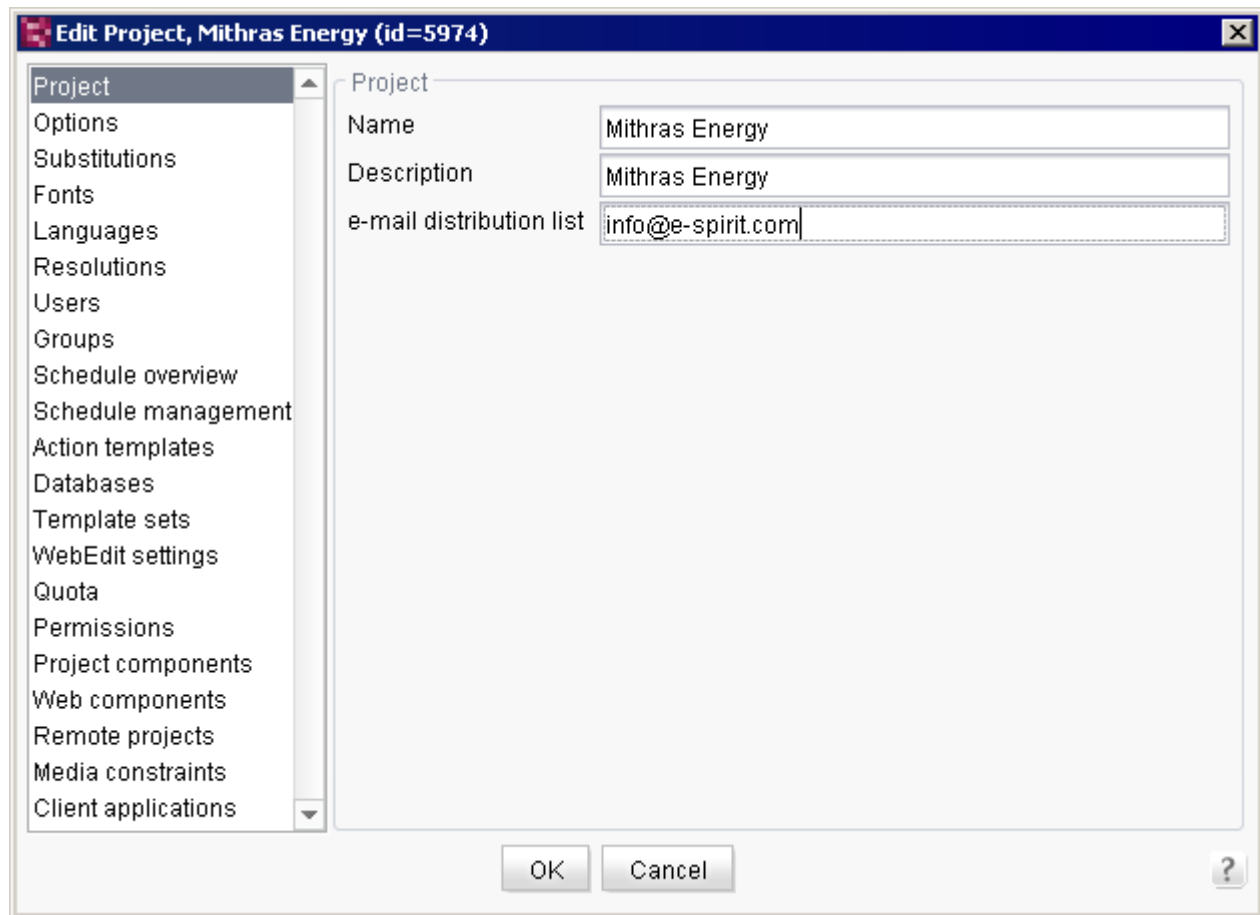


Figure 7-68: Error message when simultaneous editing is attempted



7.4.1 Project



The screenshot shows a web-based interface for editing project properties. The title bar of the dialog is "Edit Project, Mithras Energy (id=5974)". On the left is a vertical menu with the following items: 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, and Client applications. The "Project" item is highlighted. The main content area is titled "Project" and contains three text input fields: "Name" with the value "Mithras Energy", "Description" with the value "Mithras Energy", and "e-mail distribution list" with the value "info@e-spirit.com". At the bottom of the dialog are three buttons: "OK", "Cancel", and a help icon (a question mark inside a circle).

Figure 7-69: Project properties – Project

Name: the unique project name, which can also be changed as necessary, is displayed in this field. If the project name is changed, the "OK" button remains inactive until a unique name has been entered.

Description: the project description, which can also be changed as necessary, is displayed in this field. The description does not have to be unique, but it is also a mandatory field.

e-mail distribution list: after each generation procedure, an e-mail with status information is sent to the addresses entered here. A semicolon is used to separate multiple addresses.



7.4.2 Options

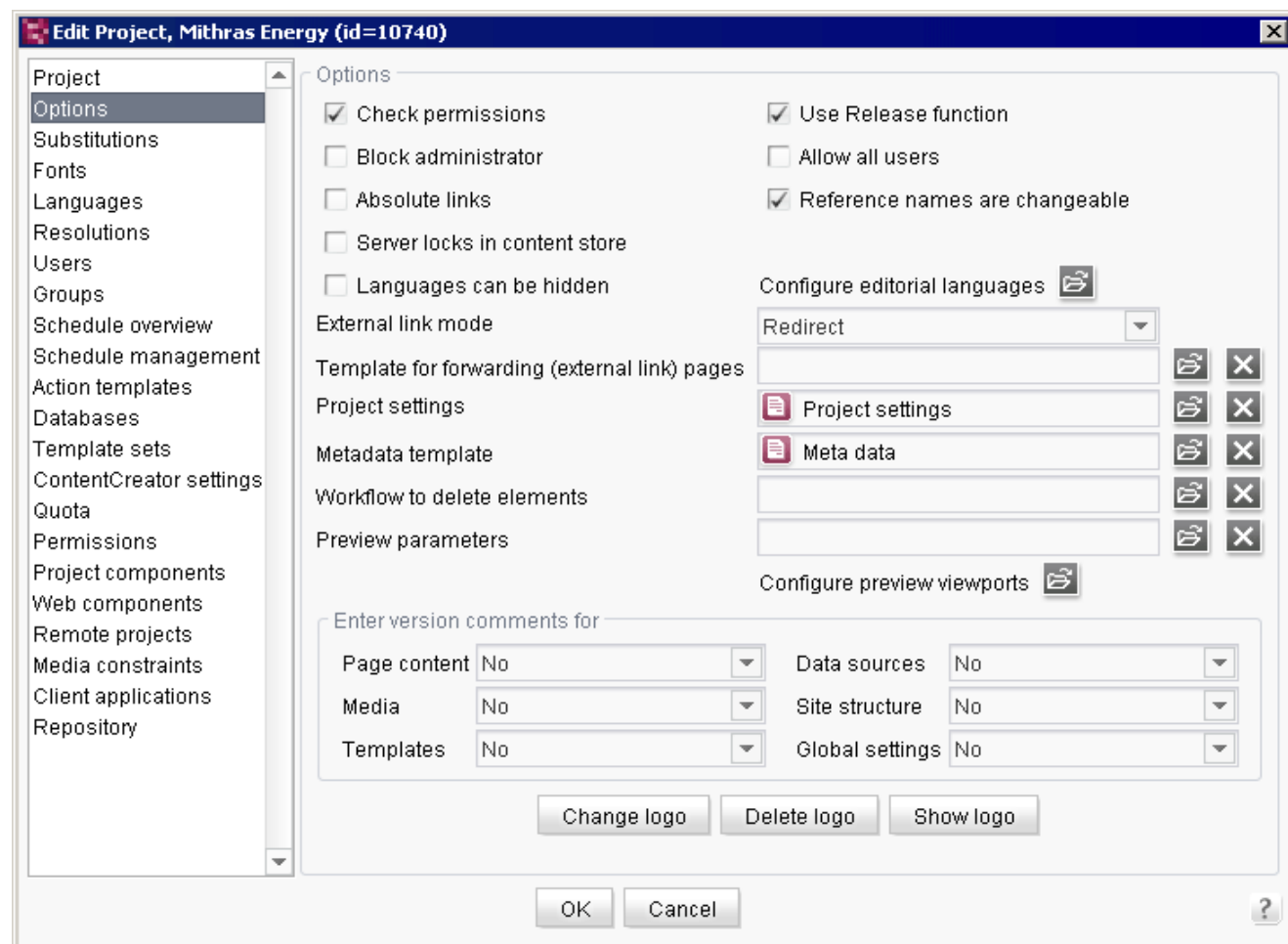


Figure 7-70: Project properties – Options

Check permissions: this option is used to define whether permissions should be checked for the project. If the checkbox is *unchecked*, the user will automatically full access to the project. You can access menu functions or objects for which access is usually only given to project administrators. If the checkbox is *selected*, the user's permissions are evaluated. A user who is not a project administrator, for instance, cannot generate a project. This option is selected by default when a project is first created.

Use Release function: if this checkbox is *selected*, FirstSpirit makes a distinction between the released and current project status. As soon as a change is made in the project, the change must be approved for release by a user with the appropriate permissions (e.g. by the "editor-in-chief"). A project status that is not approved for release, such as a page from the page store, is not copied during the next generation task. (The last released status of the page will be included



instead.) If the checkbox is *unchecked*, every change made to a project is copied automatically to the release state ("autorelease") and then takes immediate effect.

ContentCreator 5 does not support projects that do not use the Release function.

If the option is *selected again*, a query appears: "Do you want to create a release version of all stores?" Clicking "Yes" generates a release version of all stores with the exception of the template store. This procedure may take some time to complete. Clicking "No" activates the release option without creating a release version of the stores. This option is selected by default when a project is first created.



Unchecking this option only makes sense for small projects and has negative consequences on the system's overall performance, since an automatic release occurs every time a change is made.

Block administrator: if this checkbox is *selected*, access to this project is blocked for the administrator who is created automatically during the initial installation of a FirstSpirit Server (user ID 1, login: Admin, see 7.1 page 213). This administrator no longer sees the project in the SiteArchitect or ContentCreator project selection list and can no longer edit it. The server administrator tasks can, however, still be viewed:

- Create new/export/delete projects
- Add user
- Change settings for all projects
- Define project administrators
- Install and remove editor and function components
- Perform specific server operations

If this option is *unchecked*, then the server administrator can also view the tasks of a project administrator and select the project via the project selection list in SiteArchitect and ContentCreator.



Allow all users: if this checkbox is *selected*, all users known on the server automatically receive access to the project. This means even users who were not explicitly added as users of the project and thus were not listed as a "user" under the project properties as well as users who have been authenticated by the server (e.g. via LDAP, SSO, etc.) will have access to the project (see 7.4.7 page 313). These users are not part of any internal project group and have only limited permissions in the project if they are not members of an external project group. These permissions can be configured in the "Everyone" project group access permissions. Access to the project by "external users" can be limited by providing access to the project for members of an external group instead of selecting "Allow all users". Even these users no longer have to be explicitly added to the project (for more information on external groups, see Chapter 7.4.8.2 page 317). If the checkbox is *unchecked*, the users who are to have access to the project must be explicitly added to the project (see Chapter 7.4.7 page 313).

Absolute links: absolute paths can also be generated for the page references instead of the relative paths. This project property is used to influence all references on pages. The required prefix for completing the absolute links is read out from the field "Prefix for absolute paths" from the generation task (see Chapter 7.5.10.2 page 406). This option does not refer to media references. If media references are to be displayed using absolute paths, the reference must include the attribute "abs". When setting "abs:2", the prefix for links to media and pages is not used (see the FirstSpirit Online Documentation).

Reference names are changeable: if this option is selected, the reference names can be changed in the relevant project. If the checkbox is unchecked, the reference names cannot be changed and the "Change reference names" menu item is grayed out. By default, the option to create new projects is selected so that reference names can be changed as usual. Regardless of the setting in the project properties, server and project administrators can change reference names any time.



The "Rename" function (F9) in SiteArchitect is not affected by the configuration of the "Reference names are changeable" option in the project properties: the reference names of elements without a UID (e.g. sections in the page store) can continue to be changed using "Rename". Changes to section reference names can, however, be prevented by disabling the option "Display reference names" under "Configure editorial languages" in the "Options" area of the project properties. In this case, only BeanShell (API) can be used to make changes.



Server locks in content store: if this checkbox is selected, data sources can only be edited by a single user. Other users will not be able to edit a data record in the data source. If the checkbox is unchecked, multi-user mode is in use for data sources, but this mode could lead to conflicts if multiple editors attempt to edit the same data record. The option is disabled by default. If the option is enabled, it will affect all data sources in the particular project in SiteArchitect and ContentCreator.

To learn about the consequences in SiteArchitect, also refer to the FirstSpirit Documentation for SiteArchitect, "Data entry".

Languages can be hidden: if this checkbox is selected, individual users can customize their settings to hide individual project languages in the SiteArchitect "Visible project languages" under the "View" menu. This allows the user to view only the languages relevant to him. If the user hides a language, the relevant language tab will not longer be displayed, but the content for this language will continue to be loaded.



No mandatory fields are allowed in forms that prevent the ability to save the data (e.g. "allowEmpty") because this could otherwise cause errors to occur when validating the fields in the hidden language channel. Changes can therefore not be saved in a language.



Hiding a language may only take effect after SiteArchitect has been restarted. When editing mode is active, a language tab will be hidden only after exiting editing mode and then refreshing the element (<F5>).



Configure editorial languages: clicking on the folder icon next to the "Configure editorial languages" entry opens a configuration dialog:

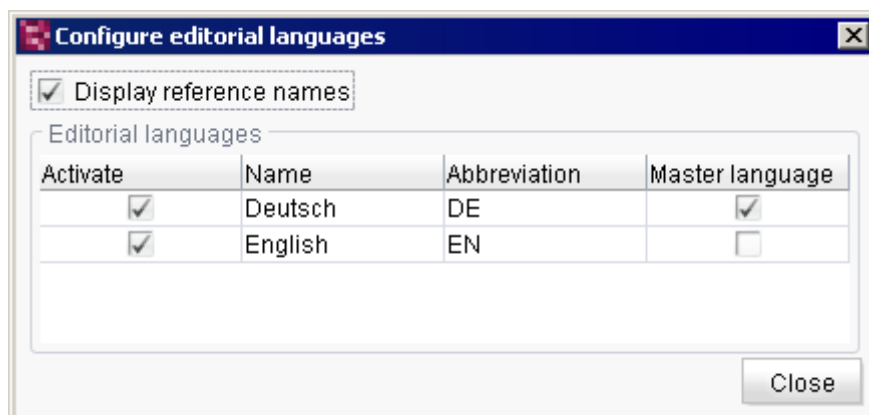


Figure 7-71: Configuring editorial languages

Display reference names: if this checkbox is selected (default setting), in addition to displaying language-dependent display names (see "Activate"), language-independent reference names of objects are also displayed.

Activate: selecting this checkbox allows the user to specify a project's editorial languages. The "active" languages can then be set in the FirstSpirit SiteArchitect as the "Preferred display language" (via the "Extras" menu). The editorial languages affect language-dependent content defined by the template developer, e.g. within the page or section templates. The relevant language-dependent labeling in the form area, for instance (labeling of input fields, tool tips, combo box elements, etc.), are displayed to the editor and also affect how objects are displayed in the tree. The editorial languages are not to be confused with the project languages (see Chapter 7.4.5 page 307).

Name / Abbreviation: the name and abbreviated form of the desired editorial language.

Master language: if this checkbox is selected, the language in this column is considered the project master language. This language is selected as the editorial language by default.

External link mode: selecting this option is relevant if the "external address (URL)" link is to be used in a project's site store. Page references can be linked directly to an external site; in this case, either "forwarding" takes place directly to the page (without storing the page on FirstSpirit Server), or forwarding takes place in addition to generating the page and storing it on the server.

Project settings: here a page template can be selected from the project which is to be used as the basis for managing the global project settings. The GUI elements defined on the "Form" tab



of the page template are displayed in the "Global settings" area of the "Project settings" and can be populated with content. The Generation and Preview context treats the contents like structure variables defined on the site store root node. Preparing the data is handled by the page template, where the functions required to manipulate project configuration data are called centrally. In the template, calculated values can be added to the context, if necessary, using the `$CMS_SET(..)$` or `CMS_HEADER` function and can thus be called by any template.

Metadata template: a template to be used for the project's metadata can be selected in this field.



Unlike selecting page templates in the SiteArchitect, there are no limitations when selecting the template in this area (for the settings page as well as for the metadata template). All available page templates are always displayed for selection—including templates that have been marked "not visible".

Workflow to delete elements: to delete elements in the FirstSpirit SiteArchitect and in the FirstSpirit ContentCreator, a project-specific workflow can be created and tied directly to the existing controls (buttons on the menu bar, context menu entry) of elements. Instead of simply deleting an object, such as a page, more complex deletion functionality can be made available via the workflow, for example, the additional deletion of dependent objects on a page.

For more information on using a workflow to delete elements, see FirstSpirit Online Documentation.

Preview parameters: In addition to the display sizes for the various output devices, other aspects can also be simulated as a preview, e.g. user-specific or role-specific perspectives. The user can simply click to display the page content as it is seen by specific user groups, e.g. private customers, partners, or business customers.

The configuration is carried out using a page template in the relevant project. This must be selected in the "Preview parameters" field. The input components defined in the page template are displayed in ContentCreator and can be filled in by the editor. The values entered by the editor can be used for the output of the current preview page.

The following input components are available for this purpose:

- `CMS_INPUT_CHECKBOX`
- `CMS_INPUT_COMBOBOX`
- `CMS_INPUT_RADIOBUTTON`
- `CMS_INPUT_TOGGLE`
- `CMS_INPUT_TEXT`



- FS_BUTTON

- **Evaluation in the FirstSpirit template**

Evaluating within the preview calculation is not possible because this is carried out for a whole project and not for one user. The variables can be evaluated using JSP code or JavaScript.

- **Evaluation during runtime (JSP)**

In the front-end server the input values can be output, for instance, using JSP code:

```
<%= session.getAttribute("fs.preview.role").toString() %>
```


In this example, `role` is the variable name of the input component the editor uses to select the role of the website user (see example above).

- **Evaluation during runtime (JavaScript)**

The input values can be output via:

```
WE_API.Preview.getParameter("role");
```

See FirstSpirit Developer API, `de.espirit.firstspirit.webedit.client.api` package, `Preview` interface for more information. This package provides methods for settings parameter too

Configure preview viewports: A range of previews can be configured for ContentCreator for the current project here. Clicking the  icon opens the following dialog:

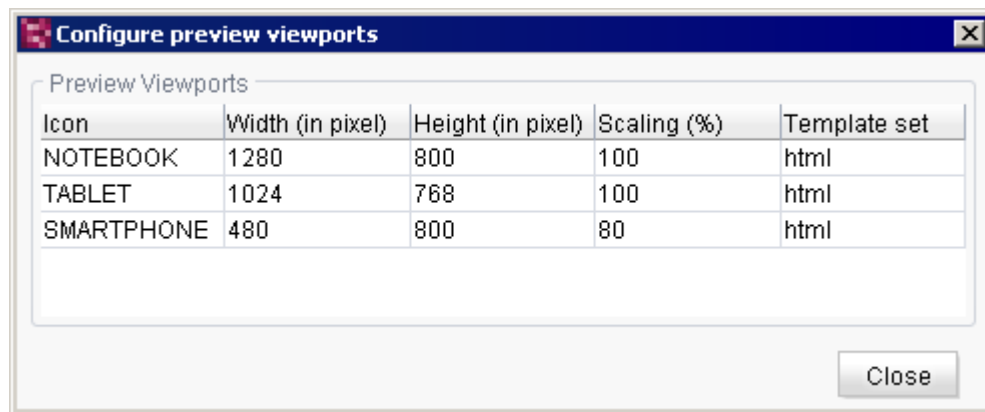






Figure 7-72: Project properties – Configuring previews

The three viewports "NOTEBOOK", "TABLET" and "SMARTPHONE" with the values shown in Figure 7-72 are pre-configured by default. Moreover, the viewport "DESKTOP" can be configured as well as further viewports with other values.



Icon: Specified identifier for the view. These identifiers are assigned icons which are displayed in ContentCreator and which allow the editor to identify the desired display size for checking the website content:

- DESKTOP: 
- NOTEBOOK: 
- TABLET: 
- SMARTPHONE: 

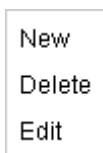
Width (in pixel): Width of the view in pixels. The default setting for the "DESKTOP" view is 1280 pixels.

Height (in pixel): Height of the view in pixels. The default setting for the "DESKTOP" view is 720 pixels.

Scaling (%): The preview can be scaled in the four available views in ContentCreator. The value specified here defines the scaling with which the relevant view is to be displayed initially. However, the editor is able to change the scaling. The default scaling for the "DESKTOP" view is 100%. For smartphones, on the other hand, 80% scaling provides a more realistic display.

Template set: From this drop-down list, the user can select a template set that is available for the project and is taken into account for the relevant view. By default, the template set which is selected for the ContentCreator (see Chapter 7.4.14 page 330), otherwise the first in the area "Template sets" (see Chapter 7.4.13 page 326).

Right-clicking with the mouse on the overview window opens the context menu:



The "New" context menu entry is used to add one of the four available views to the project. The window that opens displays the "DESKTOP" view with the default values (which are described above):



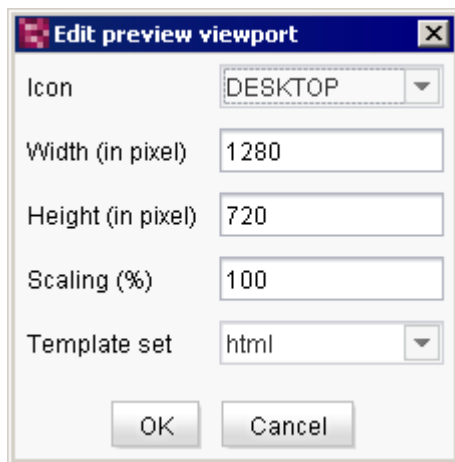


Figure 7-73: Project properties – Selecting a preview

Icon: Other views can be selected from this drop-down list. The values can be changed in the following fields depending on the project specification.

The following views are available:

- DESKTOP
- NOTEBOOK
- TABLET
- SMARTPHONE

The icons for the views are displayed in ContentCreator in the order in which they were created. In the case of the configuration shown in Figure 7-72, the icon for the "Desktop" view would therefore be shown on the left with the icon for the "Smartphone" view to the right of it.

The "Delete" context menu entry is used to remove the relevant view from the overview.

To edit a view, the user can click on the "Edit" context menu entry or double-click on the view. The window in Figure 7-73.

Enter version comments for: these combo boxes are used to specify for each individual store of the SiteArchitect whether a comment prompt is to appear related to changes to objects when the editor exits editing mode.





Figure 7-74: Entering version comments

The following options are possible:

Yes, force: if this option is selected, the user will see the comment line shown in Figure 7-74. The window can be closed by clicking on **OK** only after the editor enters text in the field.

Yes, optional: if this option is selected, the user will see the comment line shown in Figure 7-74. The window can be closed by clicking on **OK** without entering any text.

No: if this option is selected, the user can exit edit mode without any comment line appearing.

If a comment is entered when an object is changed, the comment also appears in the object's version history (in Figure 7-75 "This is a comment"):

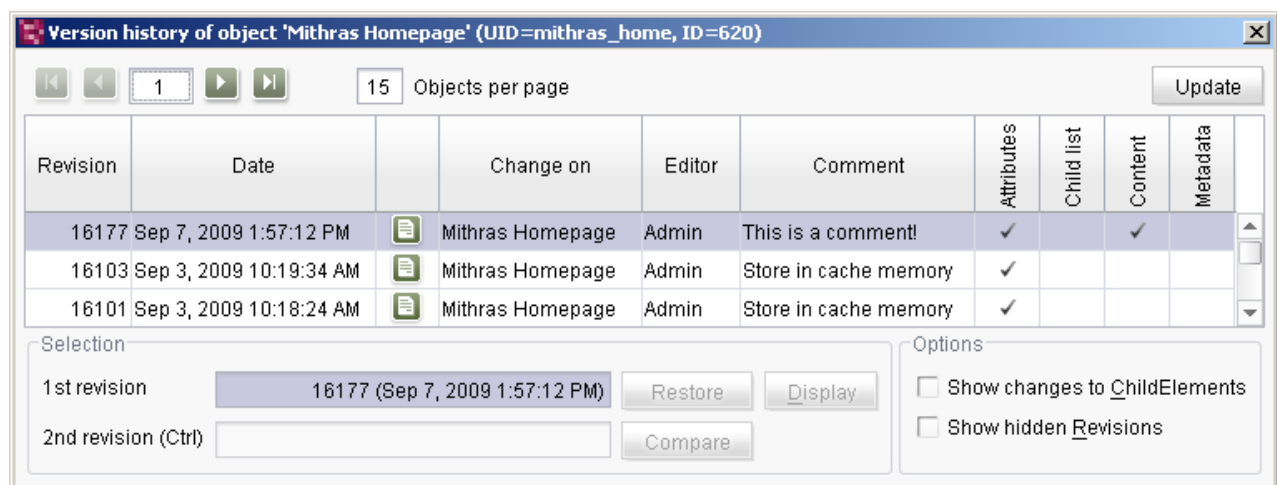


Figure 7-75: Change comment in the version history

If no comment is entered manually, a system generated comment will be stored for each revision, for example "Store in cache memory". In ContentCreator, no comments can be entered manually by default, solely system generated comments are used there.





Figure 7-76: Project properties – Editing the logo

Change logo: here you can select an image that will appear on the right-hand side of the screen (edit area) after the client is started.

Delete logo: a previously defined image is deleted. After the client is started, the FirstSpirit logo appears on the right-hand side of the screen (edit area).

Show logo: a preview of the selected logo is displayed. If no logo is included here, the FirstSpirit logo is displayed by default.

7.4.3 Substitutions

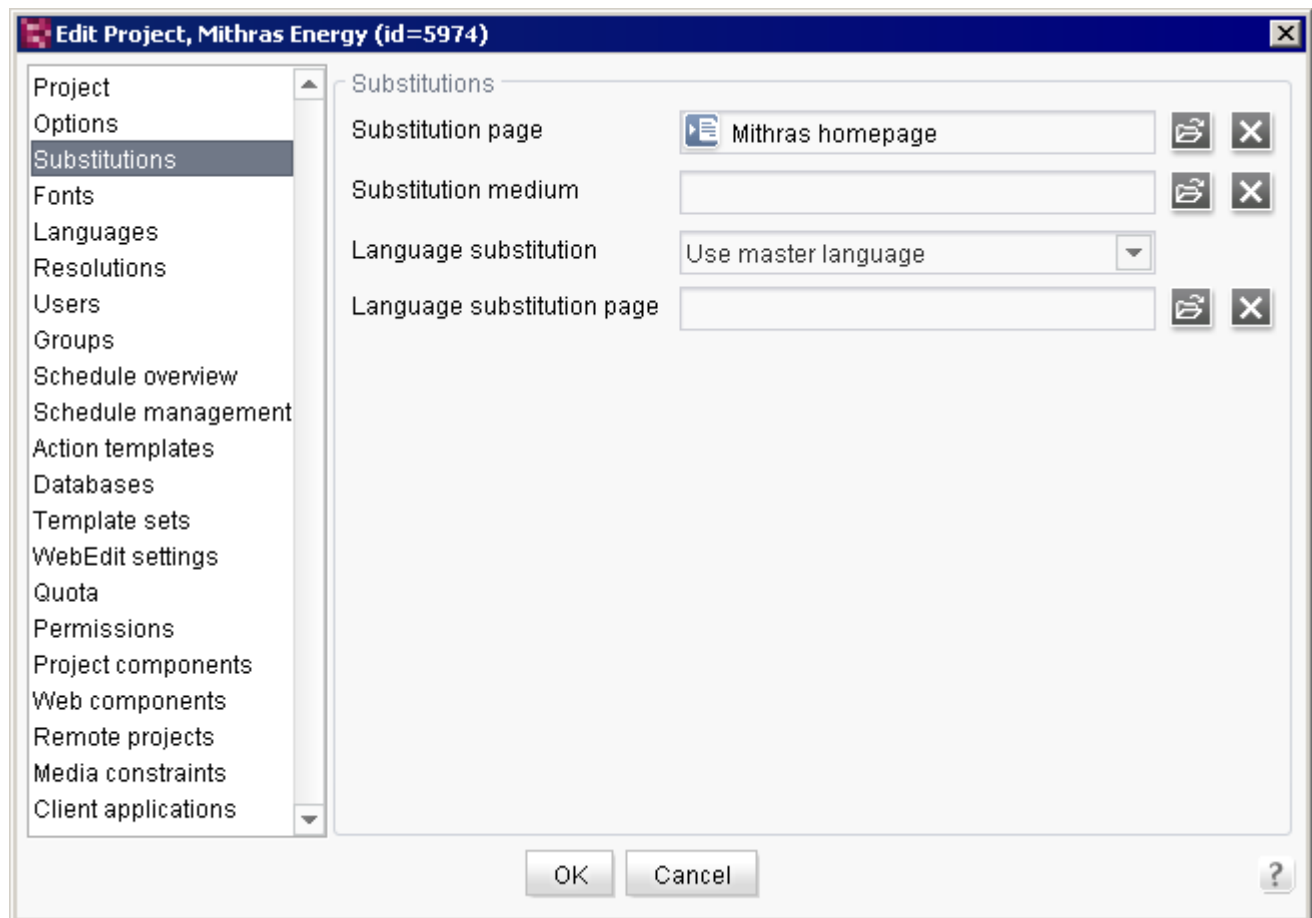


Figure 7-77: Project properties – Substitutions



Substitution page: if a page requested by a website visitor does not exist, an alternative page can be selected here for display. The file symbol is used to select the desired page from the site store.

Substitution medium: if a medium does not exist, an alternative medium can be specified here for display. The file symbol is used to select the desired medium from the media store.

Language substitution: in multilingual projects, a situation may arise where the translation of some pages is not yet available for every language. This is indicated in FirstSpirit by a corresponding check mark on the page in the page store.

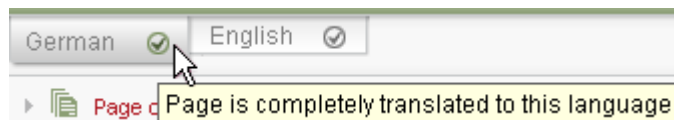


Figure 7-78: Translation in this language has been completed

In this case, a rule can be defined as to how to proceed with the corresponding pages. The following options are available:

- **Ignore:** the page is used in its current language state. The system assumes that the editor just forgot to set the check mark.
- **Use master language:** the page is generated using the project's master language instead. Since using this option can suddenly change the language when a visitor switches pages, use of this option should be considered carefully.
- **Use substitution page:** the substitution page defined above is used.
- **Use specific page:** a special language substitution page (see below) is used here.

Language substitution page: if the page actually requested has not yet been translated for a project language and the "Use specific page" language substitution rule (see below) is enabled, an alternative page from the site store can be selected for display using the folder icon.



7.4.4 Fonts

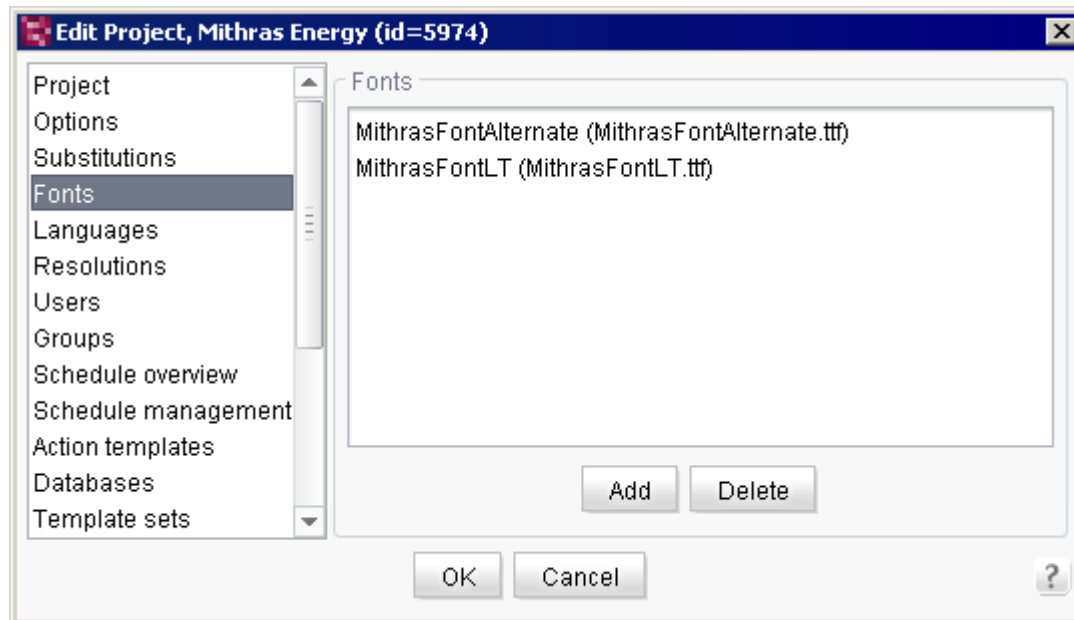


Figure 7-79: Project properties – Fonts

Here you can select the fonts to be available for the current project from the fonts installed on the server (see Chapter 7.3.4 page 250) using the "Add" button.



7.4.5 Languages

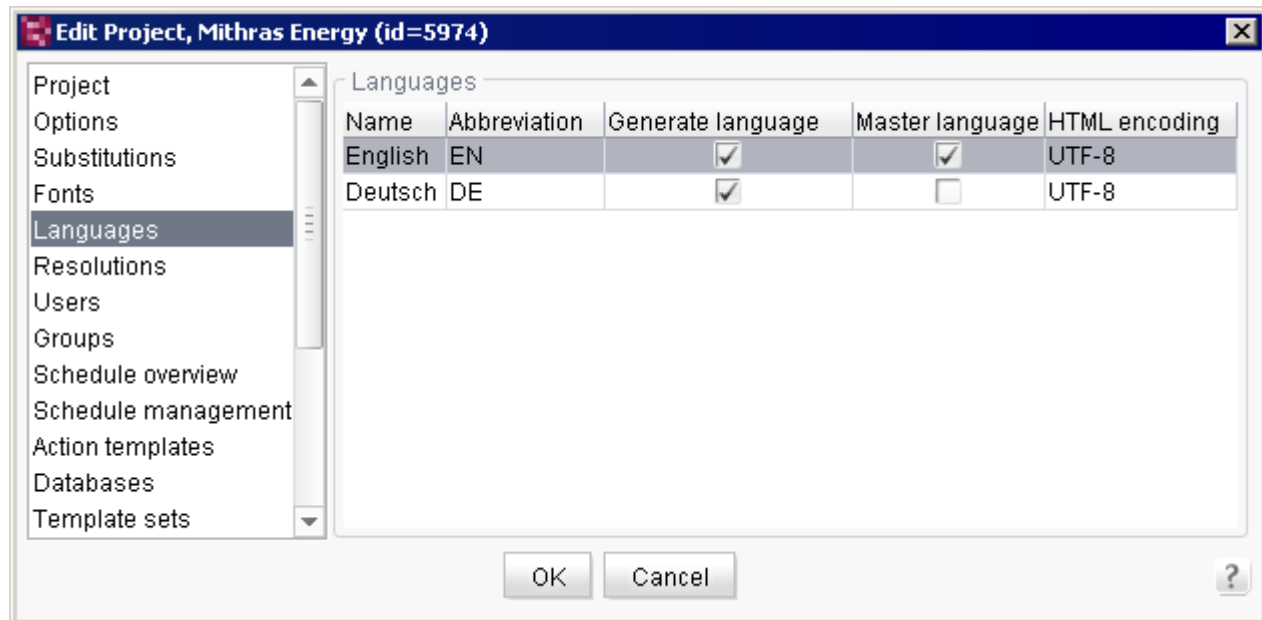


Figure 7-80: Project properties – Languages

This area contains a list of all languages used in the project. The table includes the following columns:

Name: the name displayed here is the name under which the language has been integrated in FirstSpirit. New languages first need to be made available to the server via the menu bar before they can be added to this tab for a specific project (see Chapter 7.3.6 page 257).

Abbreviation: the language code displayed is the code under which the language is used on the clients.

Generate language: if this option is not checked, the language is ignored during generation. This is practical, for instance, when the content is not yet complete in this language.

Master language: only one language can be designated as the master language. Changing the master language can be done by clicking the option for the particular language.

HTML encoding: all encodings that are supported by the particular Java version running under FirstSpirit Server are displayed in this combo box. Clicking on this field opens the combo box where the desired encoding can be selected. The default setting for each new language is ISO-8859-1, which corresponds to the Western European language region.



If a JDK version is switched to a different version that does not support (or no longer supports) the encoding, the text will appear in red. If a project is exported from one FirstSpirit server to another that does not support the encoding used in the project, a warning appears during import:

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

Right-clicking with the mouse on the overview window opens the context menu:



A screenshot of a context menu with the following options: New, Delete, Edit, Move up a position, Move down a position, and Move to last position.

Figure 7-81: Context menu (Languages area)

New: a new language can be added to a project here. A selection list of languages that are available on this server appears.



If a new language is added, all stores (page store, site store, media store and content store) are re-released in the client.



In the case of large projects with a lot of media (>>1000 or mediastore.xml file ~1MB), it is important to make sure that there is sufficient memory available for this action!

If one or more generation schedules are available for the project, you can choose if a generation is to be executed for this newly added language: "Is this language channel to be generated by all of this project's schedules?" If you select "Yes" the check marks in Figure 7-152 are set correspondingly.

Delete: use this function to remove the highlighted language from the project.





It is possible that project content could be lost when removing a language. However, a removed language can be re-added based on the corresponding template. In this case, the content previously entered in the project appears again in the corresponding language tab. However, content that was not saved at the time the language was removed cannot be recovered.

Edit: this function is used to open the highlighted language for editing. All language properties (name, ISO-639 language, ISO-3166 country, language-dependent display name) can be changed, except for the abbreviated code, which is required for unique identification of the language in the project and is not permitted to be changed.

Name	English
Abbreviation	EN
Language (ISO-639 code)	en
Country (ISO-3166 code)	US
Display name	
English (EN)	English
German (DE)	Englisch

OK Cancel

Figure 7-82: Editing languages

Language-dependent display names can be defined for a project language. The relevant input windows are displayed in the "Display name" area of the form. Language-dependent display names can be defined for all of a project's editorial languages.

Move down/up a position: the defined project languages are displayed in the defined order in SiteArchitect. Using the context menu, the language tab can be repositioned in the project. To reposition, it is only necessary to select the relevant entry and then move it up or down incrementally using the context menu or the relevant buttons. The following applies: the master language always remains in the first position.

Move to last position: this entry can be used to move the selected project language to the last position.



7.4.6 Resolutions

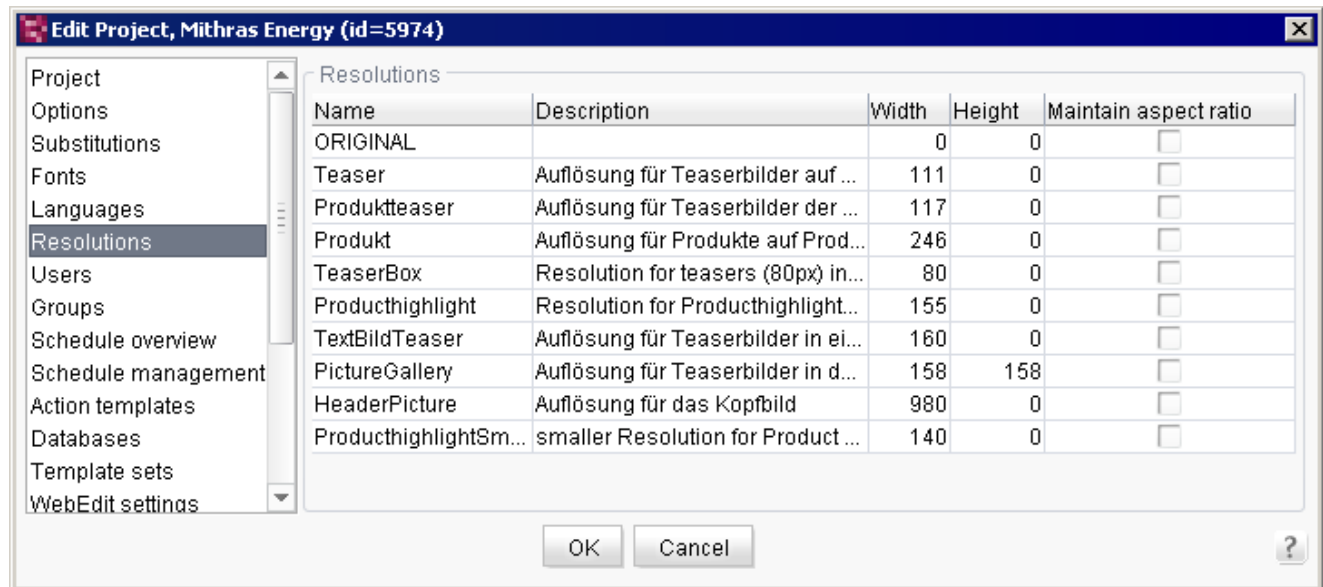


Figure 7-83: Project properties – Resolutions

This area contains a list of all resolutions defined for the project. The table includes the following columns:

Name: the unique technical name specified for a resolution is displayed here. This name is required for identification in the project and cannot be changed at a later time. A project initially only has "ORIGINAL" available for the resolution. This resolution represents the unchanged medium in the resolution in which the medium was added to the media store (symbolized by the "0" value for both width and height). When specifying a name, a distinction is not made between upper and lowercase.

Description: input field for an optional technical description of the resolution. This information is only provided within the project properties.

Width/Height: the width or height of a resolution is displayed here in pixels. A "0" means that the width or height is the result of the aspect ratio related to the original resolution. The maximum value is 5000 in each case.

Maintain aspect ratio: this option is used to specify whether the aspect ratio of the original image is to be ignored or is to be taken into account for the respective resolution when subsequently output to the website. If the option is selected, the aspect ratio of the original image is maintained.





Resolutions for which this option is disabled can be output to the website as compressed if the aspect ratio of the original image is not the same as the aspect ratio of the corresponding resolution. The option should be selected to prevent this.

Right-clicking with the mouse on the overview window opens the context menu:

- Delete
- Change
- New
- Move up a position
- Move down a position
- Move to last position

Figure 7-84: Context menu (Resolutions area)

Delete: this function is used to remove the selected resolution from the project.

Change: this function is used to open and change the highlighted resolution. The language-dependent display names and the language-dependent descriptions for a resolution can be changed as well. It will not be possible to change any other properties at a later time.

New: this function is used to add a new resolution to the project. Calling this function opens a window in which the settings for the new resolution can be modified.

In addition to the above described fields for the unique, technical name, description and width and height definition of a resolution, language-dependent display names and language-dependent descriptions for a resolution can be defined in all editorial languages.

The field **Comment** is a mandatory field and can not be modified subsequently. Text which is entered here will be shown in SiteArchitect as a tooltip for the respective resolution in the Media Store.

It is not possible to create a resolution called "preview"/"PREVIEW" or "original"/"ORIGINAL" as these are reserved, internal identifiers.

If a "preferred display language" is defined in FirstSpirit SiteArchitect, the corresponding language-dependent display names of the resolutions are displayed in SiteArchitect.



After the new resolution has been defined for media, a new resolution line appears in the Media Store.

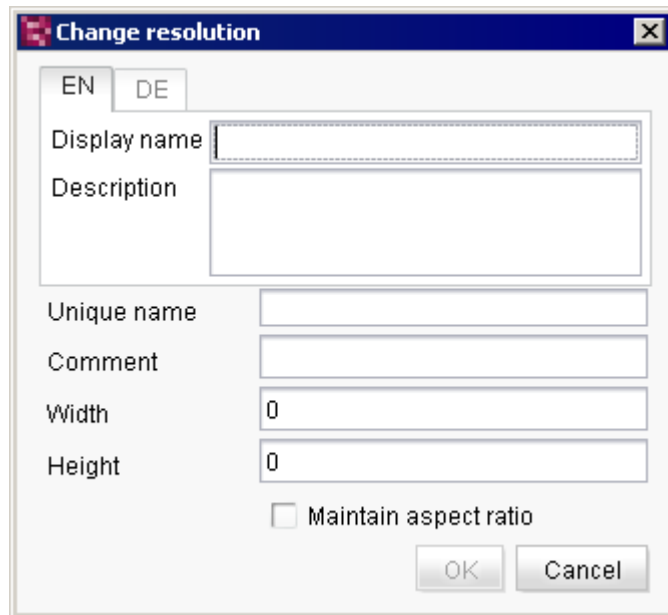


Figure 7-85: Adding a new resolution



The names of resolutions already deleted cannot be reused, since otherwise media that were calculated based on the deleted resolution could use them.

Move up/down a position: use this function to move the selected resolution up or down within the table. The order of the resolutions in this table has an effect on the representation in the Media Store in FirstSpirit SiteArchitect.

Move to last position: Use this function to move the selected resolution to the last position of the table.

In addition to the use of the context menu, the order of the resolutions can be modified by the mouse pointer (drag-and-drop) and per keyboard: Use the arrow keys to navigate through the table and move the lines with pressed Ctrl key.



7.4.7 Users

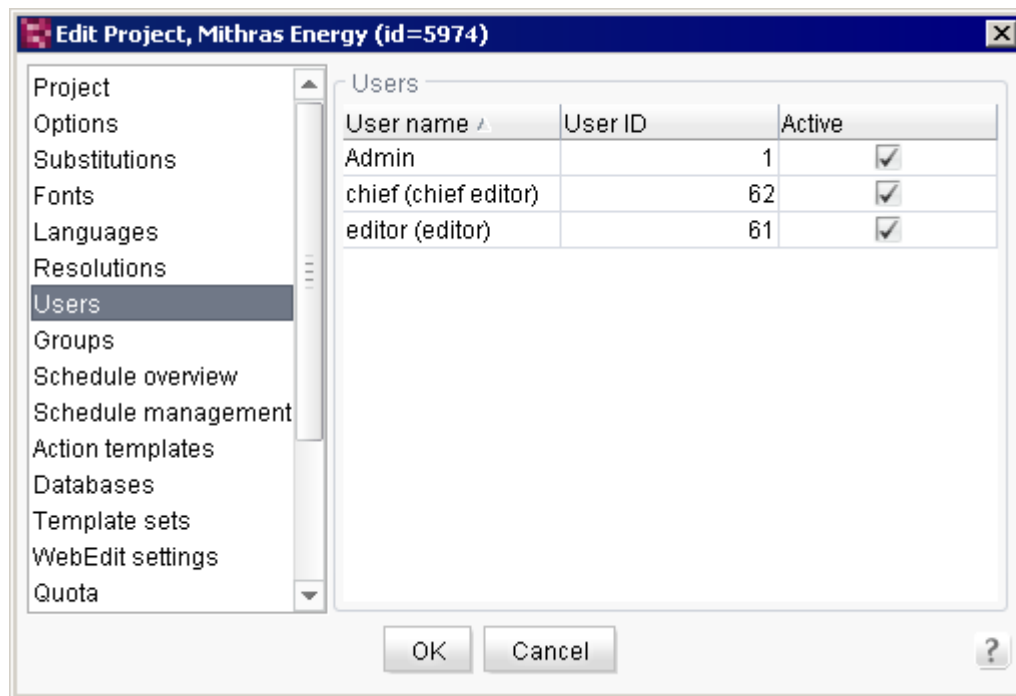


Figure 7-86: Project properties – Users

All users who have access to the project are listed here. The table includes the following columns:

User name: the unique user name used to identify the user on the server is displayed here. In the case of users who were added manually, this is the name that is entered in the "Login" field when a new user is created (see Figure 7-22). Users who are authenticated from an external system are also identified by the login and listed under the name here. If an optional entry was defined for the user, this entry is displayed in the list in parentheses next to the login.

User ID: the ID is specified automatically by the system and cannot be changed.

Right-clicking with the mouse on the overview window opens the context menu:



Figure 7-87: Context menu (Users area)



Advanced functions, which will be explained in the next Chapters, can be used via the context menu:

7.4.7.1 Deleting users from the project

Use this function to remove the highlighted user from the project. The user to be removed is selected in the user preview and then accessed via the "Delete" context menu entry. The selected user is then removed from the project.

7.4.7.2 Adding users to the project

The "Add" context menu function is used to add a new user to a project. A selection list of all users on the server appears from which the desired user can be selected. The users obtain access permissions to the project only after the assignment is made. Assignment to a project group gives the project users advanced access permissions (see Chapter 7.4.8 page 315).

FirstSpirit makes a distinction between users who were added manually to the FirstSpirit server and users who were imported automatically from an external system:

1. Users added manually: the user is added manually via the "Users" menu item of the ServerManager menu bar (see Chapter 7.2.4 page 235). A user who has been added manually to the server can be assigned to a project within the project properties (see 7.4.7 page 313) and is then automatically a member of the "Everyone" project group, but can also be added to any number of other groups (see Chapter 7.4.8.6 page 321).
2. Users added automatically: in addition to adding users manually, it is also possible to import users automatically via an external system. Users who are authenticated on the FirstSpirit server via an external system (e.g. LDAP) are automatically added to the FirstSpirit server as a user after logging in for the first time (and thus appear in the list of users) without being added explicitly via ServerManager. The assignment to groups is similar to item 1. These users can also be members of an external group (see Chapter 7.4.8.2 page 317).



7.4.8 Groups

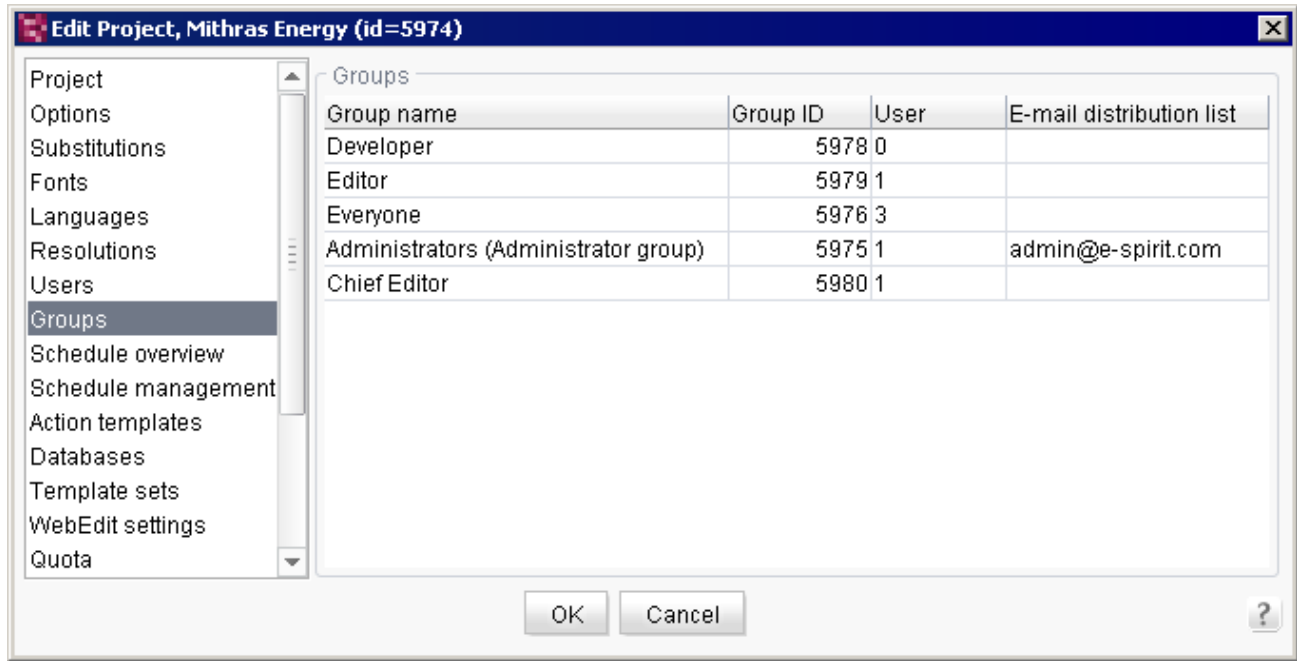


Figure 7-88: Project properties – Groups

This area lists all groups that have access to the project. Each project initially includes the default "Administrators" and "Everyone" groups, which cannot be deleted. The group "Everyone" has got the permissions "Visible" and "Read" by default within the clients, the group "Administrators" has got all permissions by default in the clients. For more information about editorial permissions see Documentation for the *FirstSpirit SiteArchitect*, Chapter "Editorial permissions".

Any number of different groups can be defined for each project. Groups contain a large number of users, but do not contain any groups.

Group name: the column shows the unique group names. Group evaluation is case sensitive. **From FirstSpirit version 5.1R2 on**, the group with the name "External Synchronisation" can be used. Users of this group have access to the functionality "External Synchronization" in SiteArchitect. For more information please refer also to *FirstSpirit Online Documentation* / "Advanced topics" / "External synchronization".

Group ID: the column shows the unique group ID that is automatically assigned to a group when it is created.

User: number of group members. Number of users added to this group. In the case of internal groups, users can be added or removed (see Chapter 7.4.7 page 313). In the case of external



groups, the number of users cannot be changed via ServerManager; instead of displaying the number of users, the note "External group" is displayed here (see Figure 7-89). (For more information on internal and external groups, see Chapter 7.4.8.2 page 317.)

E-mail distribution list: this field is used to specify e-mail distribution lists for groups to which e-mail messages are to be sent when a workflow activity or transition is carried out. This makes it possible to send e-mail messages that are sent as part of a workflow to all members of external groups.

Group name	Group ID	User	E-mail distribution list
Developer	5978 0		
Editor	5979 1		
Everyone	5976 3		
Administrators (Administrator group)	5975 1		admin@e-spirit.com
Chief Editor	5980 1		

Figure 7-89: Project properties – Groups: Group overview

Groups and access permissions: specifying and managing access permissions can be done much more easily using group definitions. For instance, a new group called "Editors A" can be defined if a certain area should be hidden from a number of editors and the number of editors changes occasionally. All editors who should not see the area are entered in this group. The root of the particular subtree in FirstSpirit SiteArchitect is hidden from the "Editors A" group by revoking the corresponding permissions for the group. If at a later date it is necessary for a particular editor to have access to the area, this user can simply be removed from the "Editors A" group, which does not require modifying the permission definition in FirstSpirit SiteArchitect.

Right-clicking with the mouse on the overview window opens the context menu:

Delete Group
Create new group
Rename group
Edit E-mail Distribution List
Show group
Remove user
Add user

Figure 7-90: Context menu (Groups area)



Advanced functions, which will be explained in subsequent Chapters, can be used via the context menu:

- Delete group (see Chapter 7.4.8.1 page 317)
- Create new group (see Chapter 7.4.8.2 page 317)
- Edit e-mail distribution list (see Chapter 7.4.8.3 page 319)
- Show group (see Chapter 7.4.8.4 page 320)
- Remove user (see Chapter 7.4.8.5 page 320)
- Add user (see Chapter 7.4.8.6 page 321)

7.4.8.1 Delete group

The "Delete group" context menu item is used to delete groups that have been added to a project. A confirmation prompt appears before explicit deletion. Once deletion of a group is confirmed, the group is removed from the project and no longer appears in the Groups area of the project properties. When the group is deleted, all members of the removed group lose their access permissions to the project (exception: users who are members of another internal or external group that is still assigned to the project).



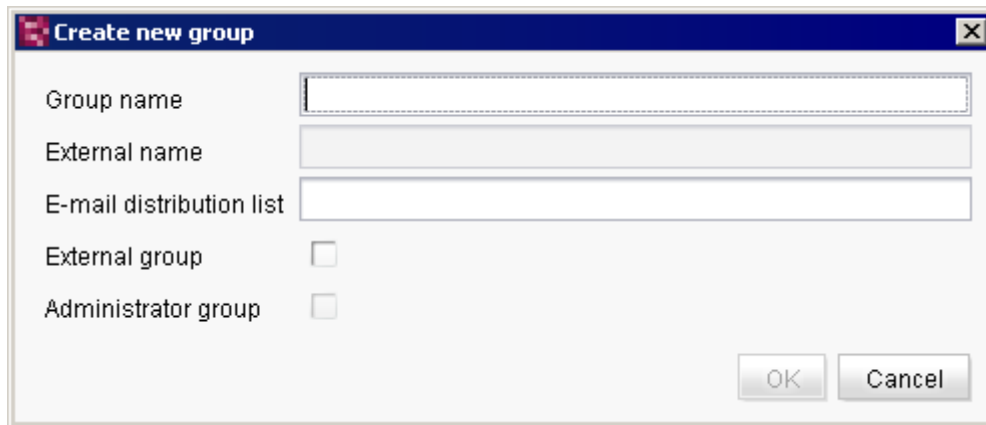
The default "Administrators" and "Everyone" groups control the initial assignment of permissions in a project and can therefore not be deleted.

7.4.8.2 Create new group

FirstSpirit makes a distinction between internal and external groups:

Internal groups are used for internal user and permission management and can be created and edited directly in FirstSpirit. For instance, users can be added to or removed from an internal group through the project properties. The properties of the group can be edited through ServerManager. To create an internal group, it is only necessary to fill in the "Group name" field (see Figure 7-91). After saving the group via the "Create new group" dialog box, the new group appears in the group overview. Once the group is created, users can be added to the new internal group (see Chapter 7.4.8.6 page 321).





The screenshot shows a standard Windows-style dialog box titled "Create new group". It features a title bar with a close button (X). The main area contains five input fields: "Group name", "External name", "E-mail distribution list", "External group" (with an unchecked checkbox), and "Administrator group" (with an unchecked checkbox). At the bottom right, there are "OK" and "Cancel" buttons.

Figure 7-91: Creating a new internal group

External groups are also assigned to a project over ServerManager, but unlike internal groups, they cannot be created through FirstSpirit; they come from a different system instead (e.g. LDAP). Membership in an external group is specified through user attributes, which means that no users can be added to an external group via the context menu (see Chapter 7.4.8.6 page 321). Users who are authenticated on the system via LDAP, for instance, receive as an attribute the membership to a group (that is not mandatorily assigned to a project) and can be added to the project via this group. The members of an external group first receive access to a project once the external group has been assigned to the project.

To add an external group to the project, the "Group name" field must be filled in first. This is the internal group name by which the group will be known and used in the FirstSpirit project. The external group name, which is the name of the group in the external system, is entered in the "External name" field. In the case of LDAP, the "external name" is LDAP-DN, e.g. `CN=Mitarbeiter,CN=Users,DC=e-spirit,DC=de`. Before the field can be edited, the "External group" checkbox must be selected.

When checking the group membership, the system checks internally whether the complete string specified for "External name" is contained in an LDAP-DN of the logged in user's groups. If, for instance, only `cn=Mitarbeiter` is entered for "External name", the group membership is adapted to the LDAP groups

`cn=Mitarbeiter,ou=Entwicklung,ou=dc=domain,dc=com`

and

`cn=Mitarbeiter,ou=Vertrieb,dc=domain,com`. To ensure the assignment is unique, the complete LDAP-DN of the group must be entered in "External name". As is generally the case with LDAP, use of upper and lowercase is ignored.



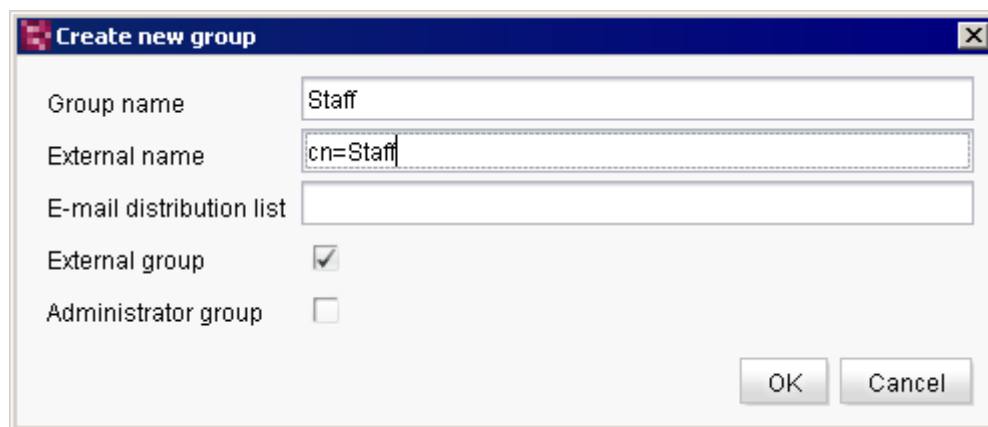


Figure 7-92: Creating a new external group



FirstSpirit does not verify whether the external group exists. If the external group name is unknown, the group is still added to the project as an external group, but in this case it has no members (assigned users).

7.4.8.3 Edit e-mail distribution list

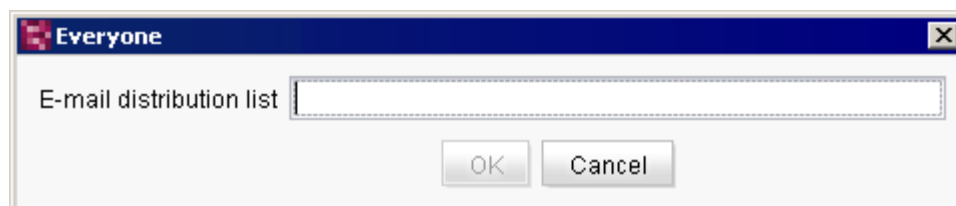


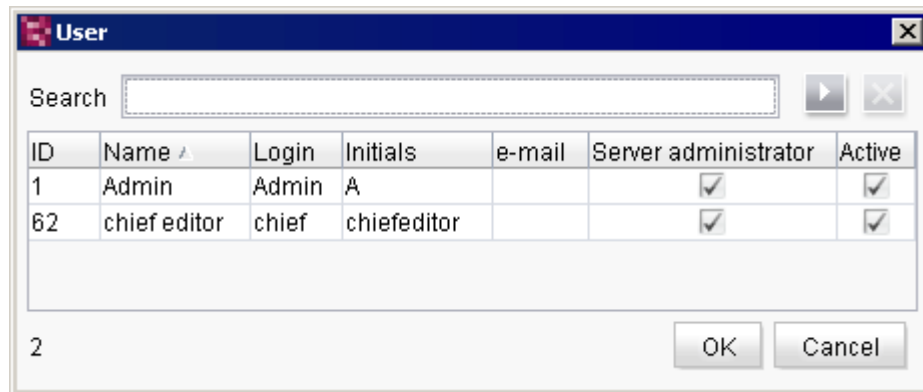
Figure 7-93: Creating an e-mail distribution list

Multiple e-mail addresses can be specified separately using a semicolon. The field can be left empty, or any e-mail addresses already in the field can be deleted.

"OK" applies the changes for the group; the e-mail distribution list now appears in the Groups overview.



7.4.8.4 Show group

**Figure 7-94: Show group**

The "Show group" context menu allows users to view a group. All users are shown here who are members of the group and thus have access to the project.

If the group is an external group, the members cannot be viewed. It is also not possible to add users to an external group or remove them from the group. Only the external name of the group can be changed here.

7.4.8.5 Remove user

A list of all members of the selected group is shown in the case of internal groups. The user to be removed is selected in the overview and then the selection is confirmed by clicking on the "OK" button. The selected user is then removed from the internal group.

It is possible to select multiple users as follows:

- Select the users from the list while simultaneously pressing the CTRL key.
- CTRL+SHIFT (selects users from a starting point to an end point)
- CTRL+A (selects all users)



The membership to an external group is assigned in the user attributes of the external system, which means that the users of an external group cannot be removed using ServerManager.





Users cannot be added to or removed from the default "Everyone" group. The following applies to this group: all users who have access to the project (via membership to an internal or external group assigned to the project) are automatically members of the "Everyone" group and receive at a minimum the access permissions defined for "Everyone".

7.4.8.6 Add user

Users can also be members of multiple groups. Each group can be assigned access permissions that can be configured separately (see Chapter 7.4.8 page 315). Members of the group receive all of the group's access permissions to the project.

In the case of internal groups, a list of all users currently being added to the project who are not currently members of the selected group is shown here. The users to be added can be selected in the overview. If you confirm the selection by clicking "OK", the selected users are added to the internal group.

It is possible to select multiple users as follows:

- Select the users from the list while simultaneously pressing the CTRL key.
- CTRL+SHIFT (selects users from a starting point to an end point)
- CTRL+A (selects all users)



The membership to an external group is assigned in the user attributes of the external system, which means that users cannot be added to an external group using ServerManager.

Users cannot be added to or removed from the default "Everyone" group. The following applies to this group: all users who have access to the project (via membership to an internal or external group assigned to the project) are automatically members of the "Everyone" group and receive at a minimum the access permissions defined for "Everyone".



7.4.9 Schedule overview

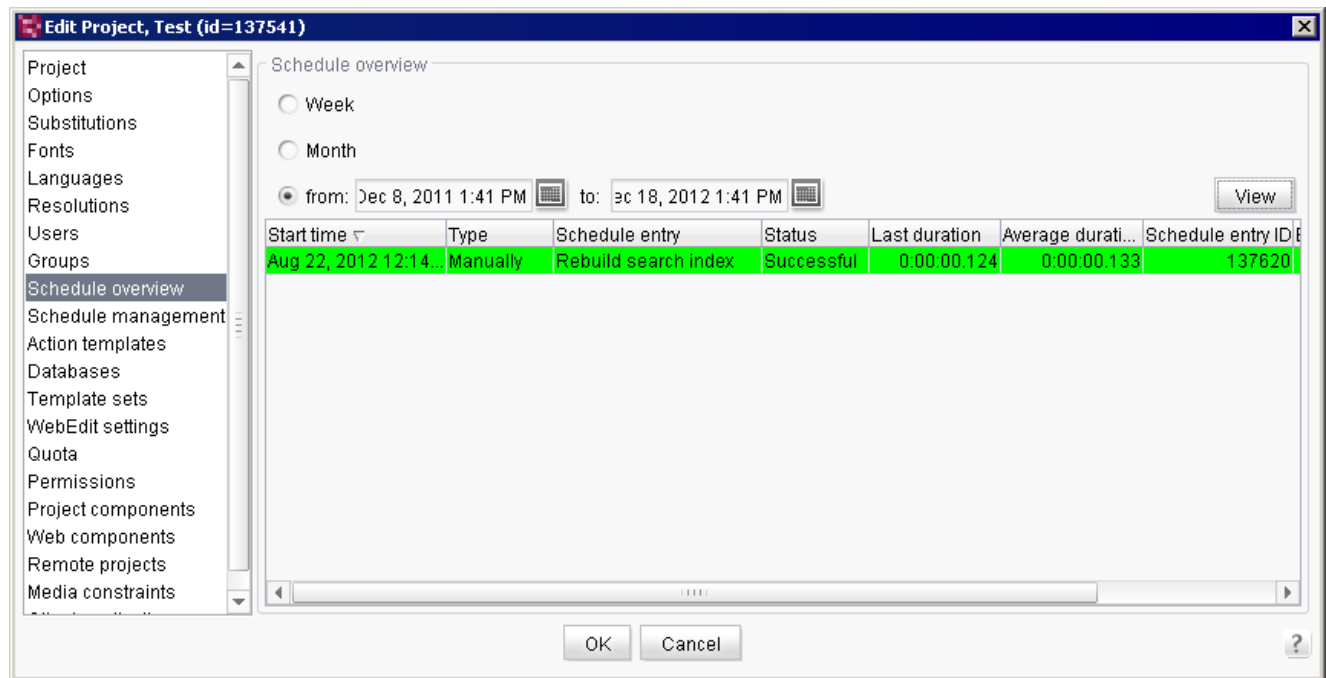


Figure 7-95: Project properties – Schedule overview

Schedule entry planning details are displayed in this area. In addition to project-based settings, server-based settings for schedule entry planning can be defined in this area (see Chapter 7.3.9 page 263). A complete schedule entry planning overview using FirstSpirit is therefore covered in a subsequent Chapter 7.5 (page 370 ff.).



7.4.10 Schedule management

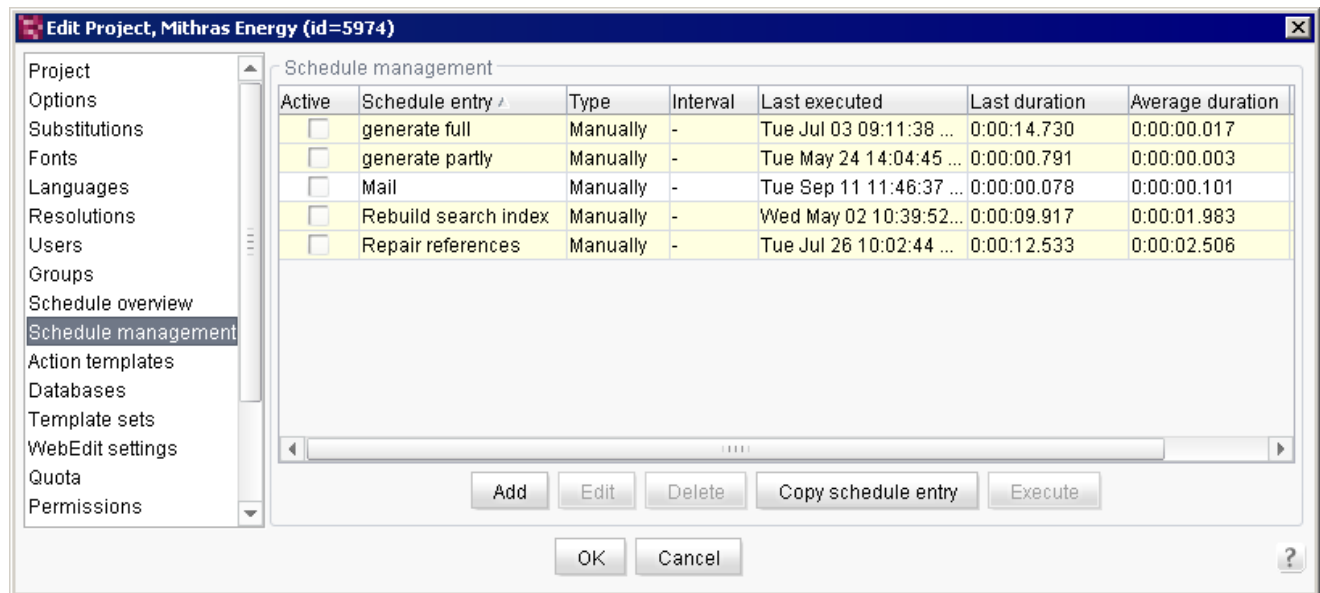


Figure 7-96: Project properties – Schedule management

Schedule entry planning provides the ability to create schedules. Schedules are used to combine advanced actions in a practical way and are either server or project based, depending on where they are created (see Chapter 7.5.2 page 375).



7.4.11 Action templates

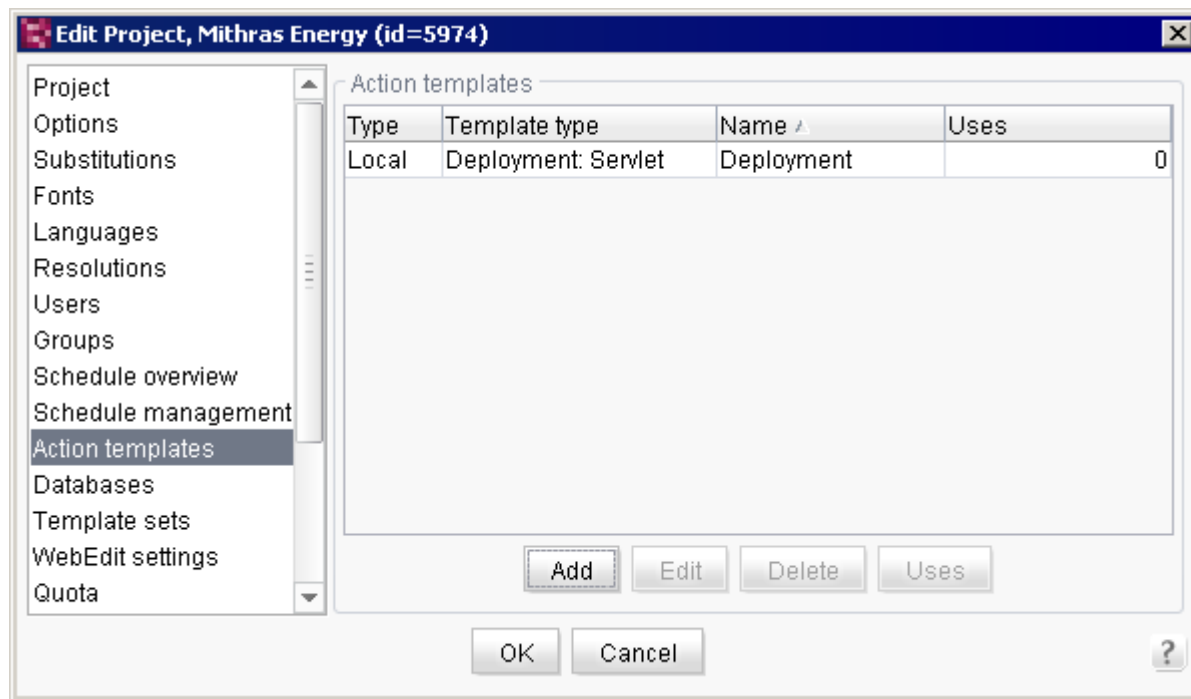


Figure 7-97: Project properties – Action templates

(See Chapter 7.5.3 page 378.)



7.4.12 Databases

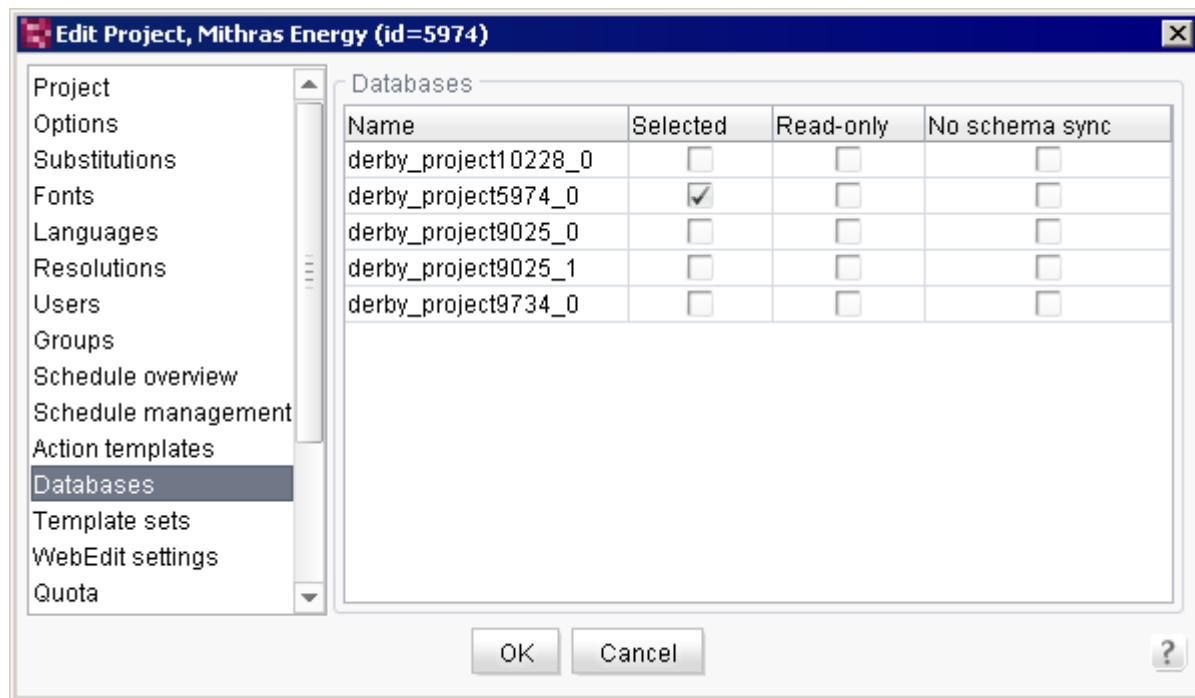


Figure 7-98: Project properties – Databases

This area contains a list of all databases defined for the project. The table includes the following columns:

Name: the names of all databases on the server are displayed here.

Selected: each database with a checkmark in this box can be used for a database schema in the FirstSpirit client. This automatically sets write access to the database for this project.

Read-only: if the selected database should only be available as read-only in the project, this option needs to be selected. In the case of external databases ("No schema sync" is selected), read-only must be selected.

No schema sync: if this option is selected, changes to the database schema that are made in FirstSpirit SiteArchitect are not made to the physical database. This setting must be selected in order to bind to external databases (see Chapter 4.9.8 page 183).



7.4.13 Template sets

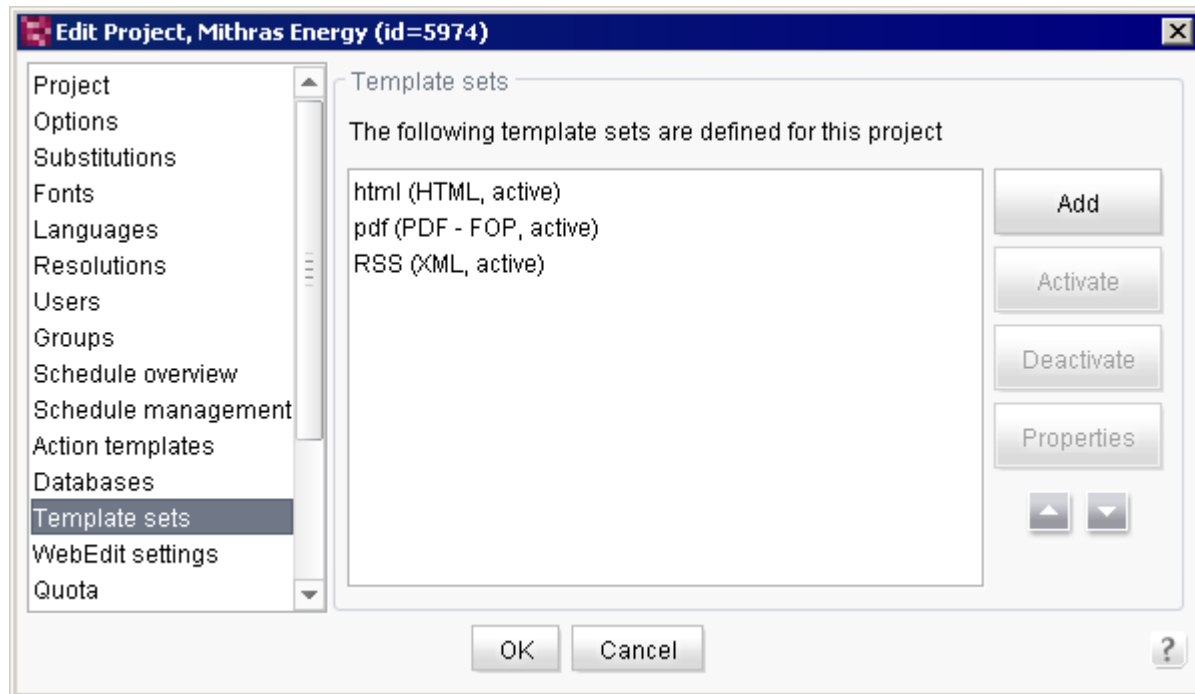


Figure 7-99: Project properties – Template sets

Template sets to be available in the project are defined in the presentation channels.

Clicking on the "Add" button opens a window where the settings for the new template set can be modified.



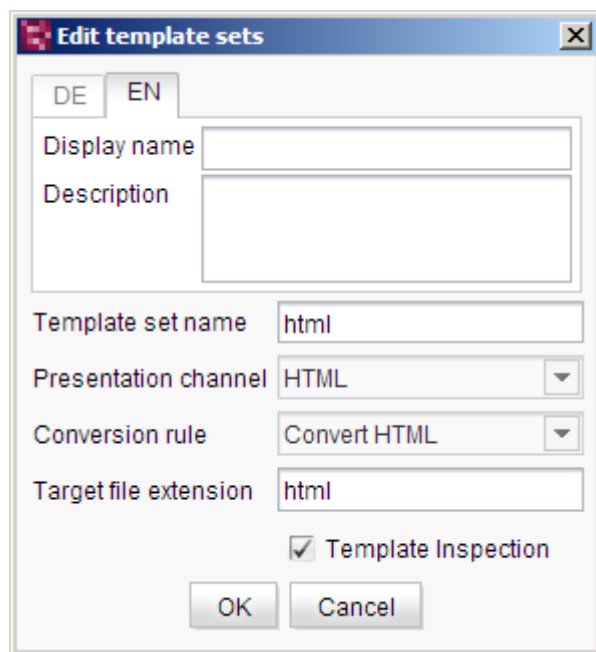


Figure 7-100: Adding a new template set

Template set name: the name of the template set that will appear in SiteArchitect for page, section and format templates is entered here.

Presentation channel: a presentation channel defined in the server properties can be selected in this field. (See Chapter 7.3.2 page 247.)

Conversion rule: a conversion rule defined in the server properties can be selected in this field. (See Chapter 7.3.3 page 248.)

Target file extension: the web server file extension is specified in this field. The linking generated by FirstSpirit to this file consists of the file name used in the site store and this extension.

Template Inspection: The source text of a presentation channel can contain XML control characters (""). If the "Template Inspection" functionality (SiteArchitect, integrated preview, "Template Inspection" context menu entry) is being used, control information is added to the tags. Depending on the (web-programming) language used, these characters may have to be quoted. However, as not all languages are able to or should be expanded, the "Template Inspection" functionality can be deactivated **from FirstSpirit-Version 5.1R3 on** for a presentation channel in the project using this option.

As an alternative, the functionality can be deactivated temporarily in the output (in the template itself) by calling:



```
$CMS_SET(#global.htmlMode, false)$
```

The Template Inspector is deactivated from this point. It can be reactivated by calling

```
$CMS_SET(#global.htmlMode, true)$
```

If template sets already exist, this option is activated by default if the target file extension is "html", "htm" or "xhtml".

For information on the "Template Inspection" functionality, also see the FirstSpirit online documentation, "Template development / Debugging / Where is the error? / Template Inspector".

Display name / Description: A language-dependent display name and a language-dependent description can be specified for each template set in all editorial languages.

If a "Preferred display language" is defined in FirstSpirit SiteArchitect, the respective language-dependent display names of the template sets are shown in SiteArchitect.

If there are one or more generation schedules, you can select if the generation is to be executed for the newly created template set: "Is this set of templates to be generated for all of this project's schedules?" Click "Yes" to set the ticks in Figure 7-152 correspondingly.



Clicking on the "Activate" button activates the selected template set. (The current status is displayed in parentheses.)

Clicking on the "Deactivate" button deactivates the selected template set. (The current status is displayed in parentheses.)

Clicking on the "Properties" button opens a window where the settings for the selected template set can be modified. (See Figure 7-100: Adding a new template set.)

Display name/Description: a language-dependent display name and description can be assigned to each template set in all editorial languages.

If a "preferred display language" is defined in FirstSpirit SiteArchitect, the corresponding language-dependent display names of the template sets are displayed in SiteArchitect.

If one or more generation schedules are available for the project, you can choose if a generation is to be executed for this newly added template set: "Is this set of templates to be generated for all of this project's schedules?" If you select "Yes" the check marks in Figure 7-152 are set correspondingly.



The template sets of a project can be re-sorted within ServerManager. To do this, the relevant entry simply needs to be selected and then moved up or down incrementally using the relevant buttons. The changed order also affects the order of tabs in SiteArchitect.



7.4.14 ContentCreator settings

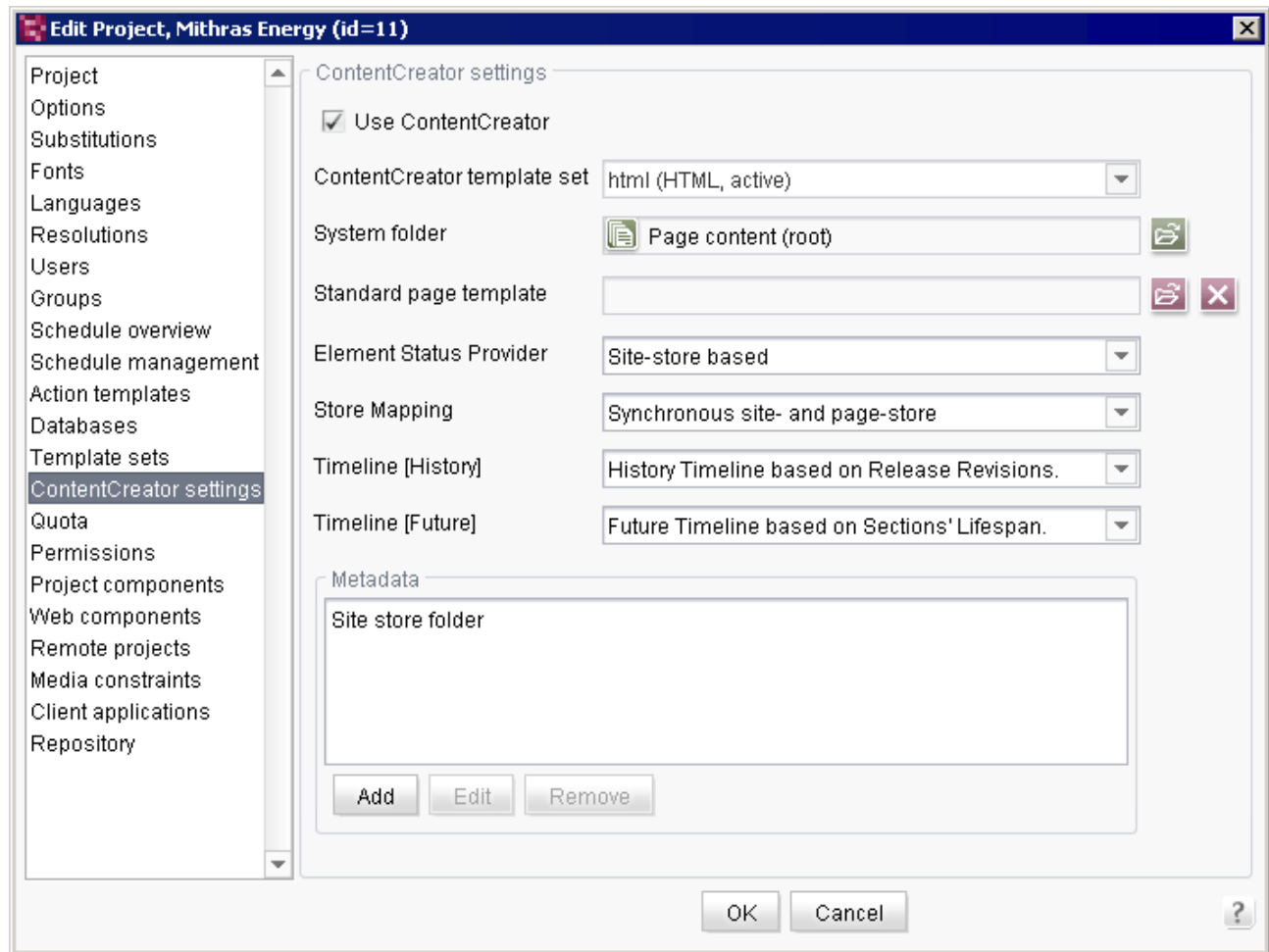


Figure 7-101: Project properties – ContentCreator settings

Use ContentCreator: if this option is selected, this project can be edited in ContentCreator (editing environment). If the option is not checked, the project can no longer be edited in ContentCreator.

ContentCreator template set: this option is used to select a template set for the ContentCreator from the template sets assigned to the project. (See Chapter 7.4.13 page 326.)

System folder: by selecting "System folder", you can specify the page store folder where new pages are to be stored. Any existing folder can be selected as the destination for the pages added to ContentCreator. Alternatively, the root node of the page store can also be selected ("Content"). This is predefined initially. Using this setting, **all** new pages are placed under the root node.



Since by default pages are added using the ContentCreator menu to a folder and page reference in the site store, for instance, the pages are also added to a folder in the page store. This folder is placed under the selected system folder.

A "Company" page with a menu item within the default setting with *Page content* as a system folder appears in ContentCreator and SiteArchitect as follows:

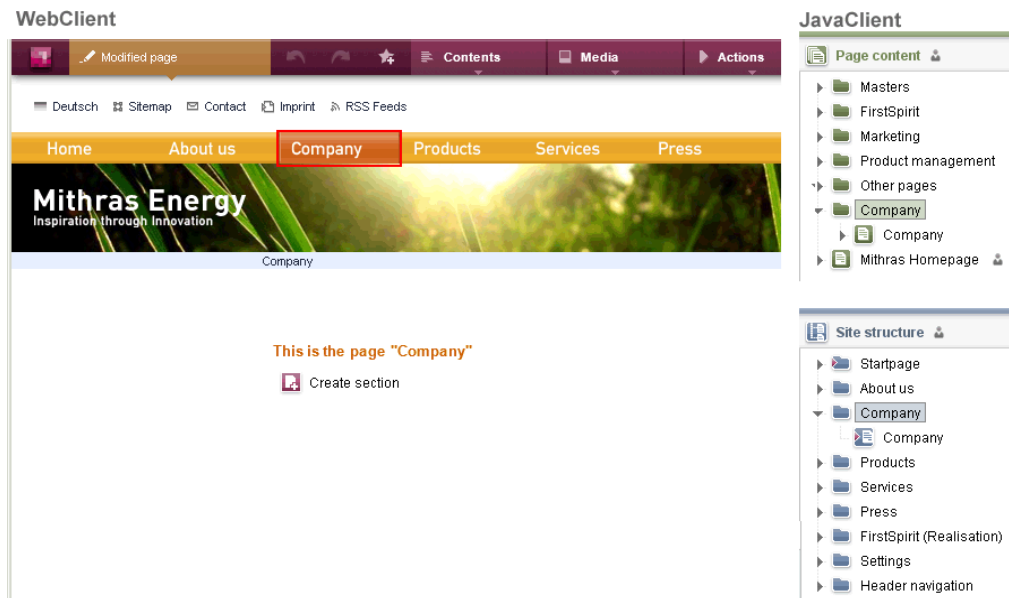


Figure 7-102: Page with menu item in ContentCreator and SiteArchitect

If the menu item in ContentCreator is removed ("Contents" / "Edit" / "Convert menu item to page"), the "Company" menu item will disappear from the navigation area and the related folder will be removed from the SiteArchitect **Site Store** and **Page Store**; the page under the selected "System folder" is retained.



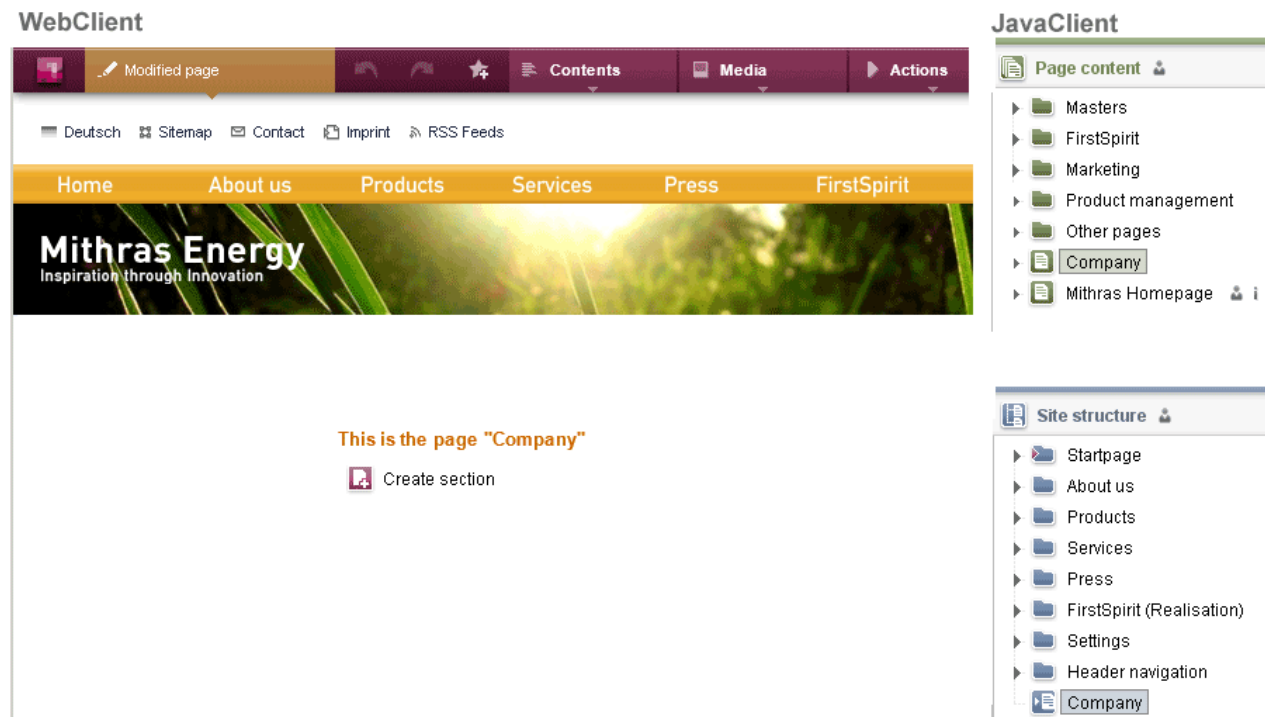


Figure 7-103: Page with menu item in ContentCreator and SiteArchitect removed

Standard page template: in this field a page template can be selected that is then preselected when creating pages in ContentCreator ("Contents" / "New" / "Create new page"). The editor then no longer has to select a template in the "Layout" area of the corresponding wizard (also refer to the Documentation for the *FirstSpirit ContentCreator*). The editor can, however, select a different template, if necessary. A standard page template cannot be selected initially. A selection affected by this can be reset to the initial state using the relevant button.

Element Status Provider: workflows are carried out by default via the page status entry in the ContentCreator menu bar (for more information on "Modified page" and "In workflow" entries, see Documentation for the *FirstSpirit ContentCreator*). The FirstSpirit object(s) to which the workflow is to refer, e.g. to pages of the site store or of the page store, can be set via the Element Status Provider. The following providers are included with FirstSpirit by default and can be selected from the drop-down menu:

- **SiteStore:** if this provider is selected, the workflows started in ContentCreator only affect pages in the site store. This means, for instance, that if a release workflow is carried out on the ContentCreator page on which the editor is currently working, after the workflow is completed, only the corresponding page in the site store will be released and not the associated menu level or the associated page in the page store. This provider is selected by default.



- **PageStore:** if this provider is selected, the workflows started in ContentCreator only affect pages in the page store. This means, for instance, that if a release workflow is carried out on the ContentCreator page on which the editor is currently working, after the workflow is completed, only the corresponding page in the page store will be released and not the associated page reference or the associated menu level in the site store. A page released with this provider might be ignored during the next generation action.

It is recommended to create a provider for each project which is customized to the respective requirements, which for example can release media or more than one object at once. See also *FirstSpirit Online Documentation*, area "Plug-In Development" / "ContentCreator Extensions" / "Interactive Features" / "Element Status and Workflow Displays". There you can find amongst others an example workflow for releasing pages including modified media.

Store Mapping: in order to add new content to FirstSpirit, a page is always needed in the page store to be used as the basis for the page. When adding new pages in the ContentCreator (via "Contents" / "New" / "New Page"), the editor can specify only the position in the site store and not the position in the page store. The "Store Mapping" drop-down menu is used to specify how and where new pages are to be stored in the page store and how renaming or moving of pages/menu levels in ContentCreator is to take place in the page store.

Depending on the project requirements and architecture, a plug-in selected from this drop-down menu identifies, for example (based on the folder and page in the site store), the appropriate folder in the page store in which the page in ContentCreator will be created, or, if necessary, creates an appropriate folder or folder hierarchy.

"SynchronousStoreMapping" is included with FirstSpirit by default. The page structure in ContentCreator matches the structure in the folder selected under "**System folder**" (see above). If a new page is created in ContentCreator, this plug-in depicts the structure/hierarchy of the site store in the "**System folder**" of the page store from the root (the site store) to the newly created page reference.

For information on developing custom plug-ins, see *FirstSpirit Online Documentation*, "Plug-In Development" / "ContentCreator Extensions" / "Management Extensions" / "Store Mapping".

Timeline [History]/[Future]: In ContentCreator, the content of the current page can be displayed at different points in time. A timeline is provided so that the editor can select a particular point in time. FirstSpirit contains default timelines for the past ("History Timeline based on Release Revisions") and for the future ("Future Timeline based on Sections' Lifespan"), but customer-specific timelines can be implemented in addition to these. Use the "Timeline" drop-down menu to define which timeline implementation is to be used for the past and which one is to be used for the future. If you want to create your own implementations, you can use the `de.espirit.firstspirit.client.plugin.timeline` package in the FirstSpirit



Developer API as a starting point. Refer also to the following section in the FirstSpirit online documentation: "Plug-In Development" / "ContentCreator Extensions" / "Interactive Features" / "Timeline Markers".

The default timeline implementation called **History Timeline based on Release Revisions** stretches back eight weeks and displays release times. The default timeline implementation called **Future Timeline based on Sections' Lifespan** displays validity periods for sections. The exact future period that is displayed by the timeline is dependent on what validity periods have been set.

You could, for example, set up your own implementations so that the history timeline displays (fixed) deployment times or tagged revisions (see FirstSpirit Access API, Interface in the `de.espirit.firstspirit.access.project` package, `createTag` method, etc.) and so that the validity of datasets is read out for the future timeline.

Metadata: extra information about FirstSpirit objects can be added using metadata. A special page template used to query and enter the desired information can also be provided to editors. The form based on the template defined for metadata is directly available in SiteArchitect (see Chapter 7.4.2 page 294, "Metadata template" for more information). In order to collect metadata via the ContentCreator as well, an entry must be created in this list for each FirstSpirit object type on which the metadata are to be collected in ContentCreator. The following FirstSpirit object types are available:

- Page
- Section
- Page reference
- Structure folder (menu level)
- Medium
- Media folder

Add: the "Add" button is used to add the desired FirstSpirit object type to the list. The desired type can be selected from the drop-down menu when it opens. For the type, an informative display name must be specified for each project language in the following dialog. This display name will appear to the editor in ContentCreator as part of the description for the metadata entry and edit menu, e.g.



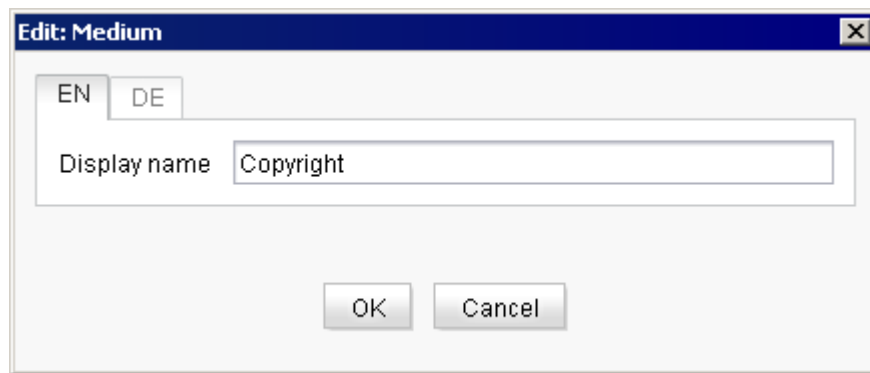


Figure 7-104: Project properties – ContentCreator metadata

The display name "Media information" selected for media in the example is displayed to the editor in ContentCreator, e.g. when uploading media to the project:

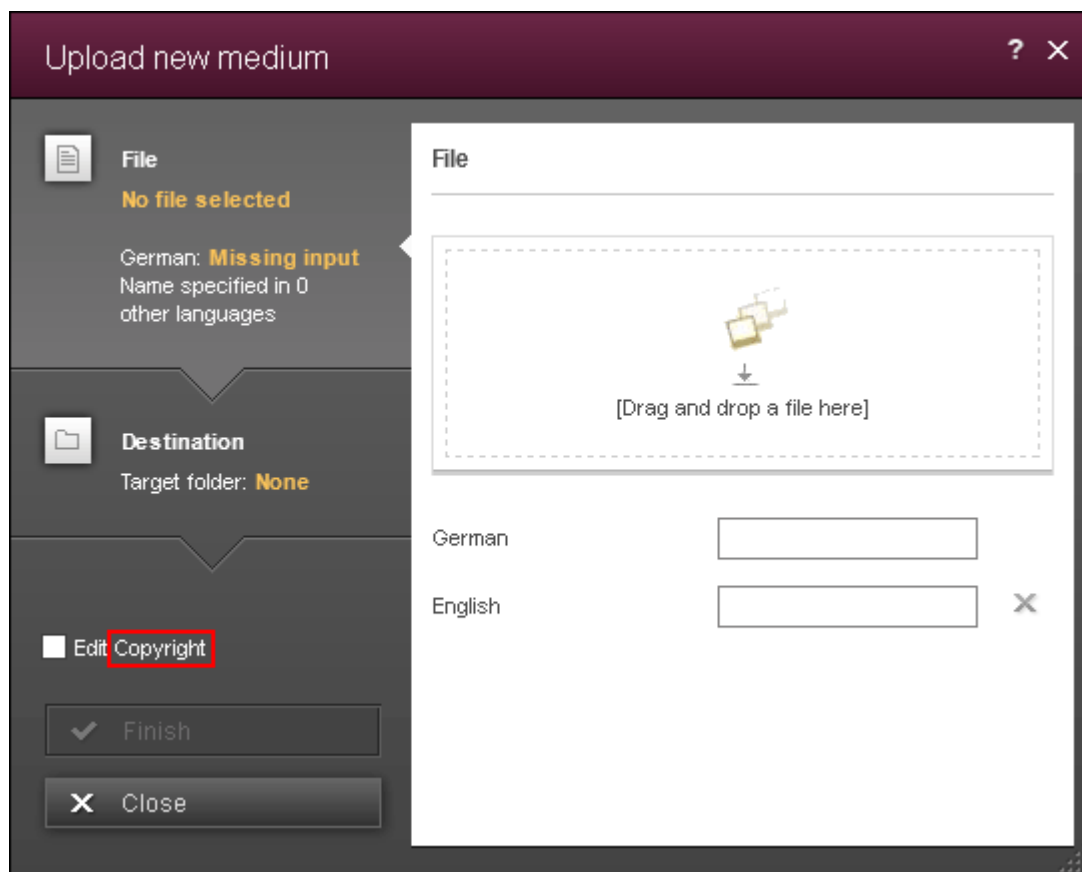


Figure 7-105: Metadata during media upload

Each type can be selected only once.

Edit: if display names already selected in one or all languages are to be changed, this can be done using the "Edit" button. The dialog box shown in Figure 7-104 opens. Only the menu name



needs to be changed in ContentCreator; this change will have no effect on previously entered metadata.

Remove: this button is used to prevent the entry of metadata in ContentCreator. If no more metadata is to be entered or if metadata is only to be entered for certain FirstSpirit object types, the types that are no longer required can be removed from the list using the "Remove" button. The corresponding menu function, (see, for instance, Figure 7-105) is then removed; this change does not affect previously entered metadata.

Report Plug-ins: The checkboxes used in FirstSpirit version 5.0 to activate report plug-ins have been removed in FirstSpirit version 5.1. In line with the general module implementation, a report plug-in is activated in FirstSpirit version 5.1 by installing the menu on the FirstSpirit Server:

- In FirstSpirit SiteArchitect, the classes included in the module are automatically reloaded once the module has been installed on the server and they are then available in the project development environment without any further configuration. If this is not desired, the report developer must disable this function in the module implementation (see *FirstSpirit online documentation*, plug-in development for more information).
- Dynamic class loading is not possible in FirstSpirit ContentCreator. The module must first be added as a web component (in the "ContentCreator" tab) and installed on the relevant web server (see Chapter 7.4.18 page 343). The report classes will then be available in the editing environment.

Note on compatibility: Modules that were developed for FirstSpirit version 5.0 should be compatible with version 5.1 in principle. As the two-stage activation of report plug-ins in version 5.0 ([1] installation and [2] activation via the checkboxes that have now been omitted) has changed with version 5.1, all modules installed in FirstSpirit version 5.1 are now immediately active (see above).



The usage of self-implemented reports in ContentCreator requires an application integration license. The related license parameter `license.APPTAB_SLOTS` applies for all FirstSpirit application integrations in ContentCreator and SiteArchitect and specifies how many reports or AppCenter applications can be used (see also Chapter 4.3.5 page 101 and Chapter 8.6.2.6 page 483). If this number is exceeded the following warning will be displayed: "This exceeds the number of licensed application integrations!" The related report icons will be shown in ContentCreator but they are inactive.

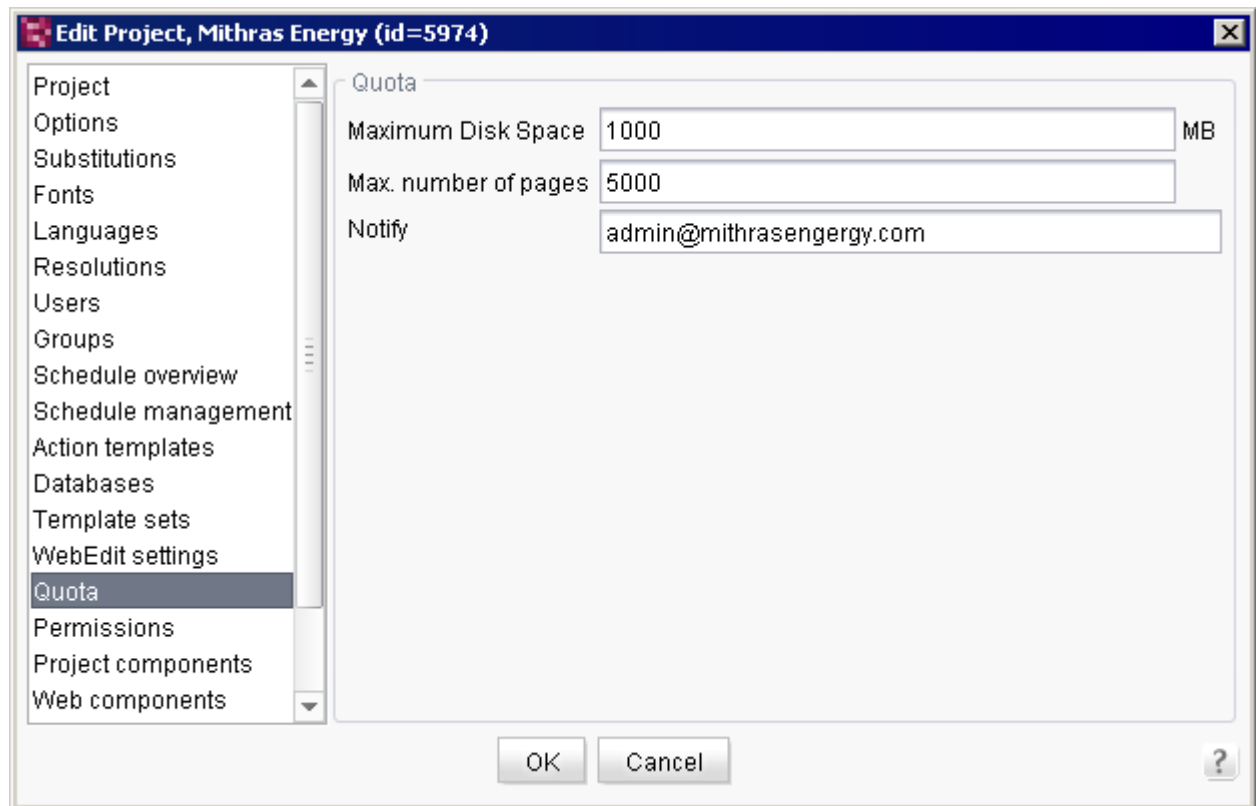




If Workflow Group Provider, Store Mapping or Report-Plug-ins are to be used in a project configuration and they were installed using FirstSpirit modules (FSM) in the FirstSpirit Server, a project-specific ContentCreator web application must be installed and activated (see Chapter 7.4.18 page 343). Web components that contain Java classes used and resources required by these plug-ins must be added to this web application. A description of how to configure these types of web components in an FSM as well as how to provide ContentCreator plug-ins specific to projects via web components can be found in the FirstSpirit Online Documentation, "Plug-In Development" / "ContentCreator Extensions" / "Implementation and Deployment".



7.4.15 Quota



Edit Project, Mithras Energy (id=5974)

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

Quota

Maximum Disk Space 1000 MB

Max. number of pages 5000

Notify admin@mithrasenergy.com

OK Cancel ?

Figure 7-106: Project properties – Quota

Maximum disk space: specifies the maximum amount of disk space to be provided on the CMS server for the project.

Max. number of pages: specifies the maximum number of pages allowed for the project.

Notify: specifies who should be notified if the disk space or page limit has been reached.



7.4.16 Permissions

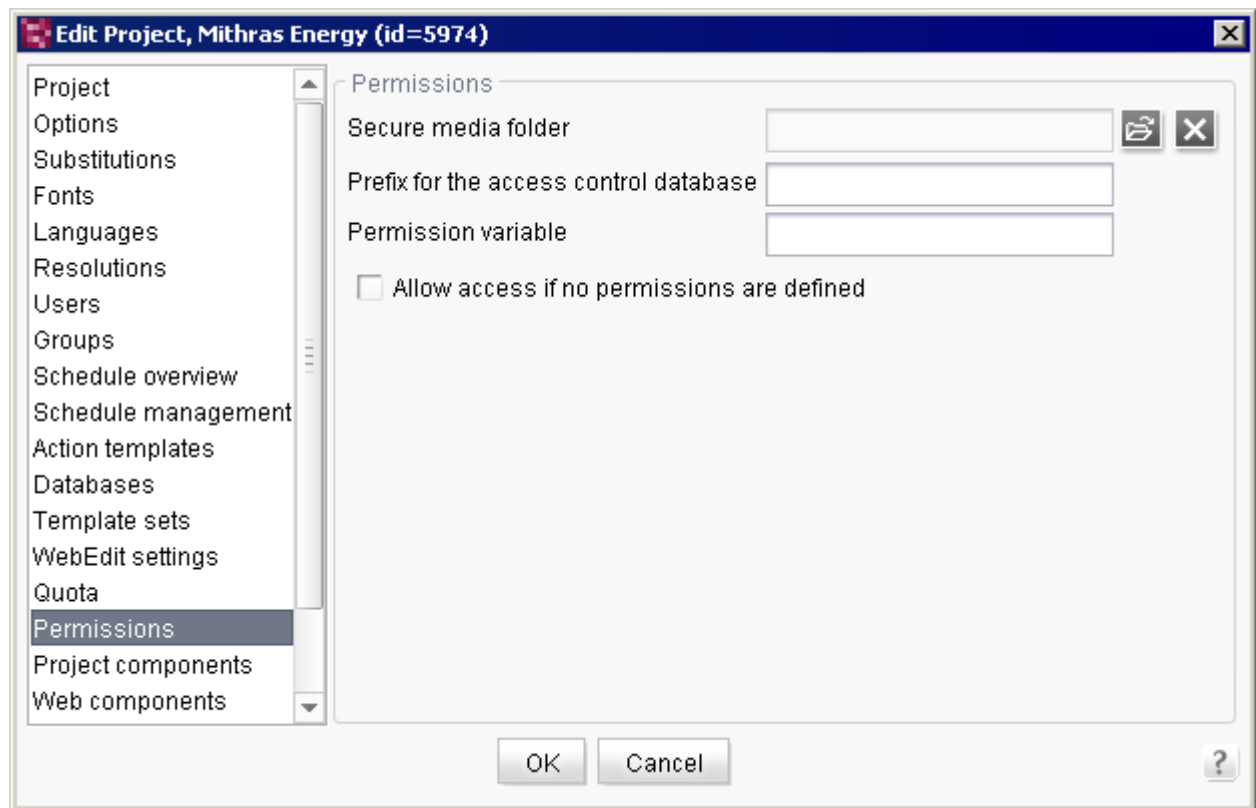


Figure 7-107: Project properties – Permissions

Secure media folder: use the folder icon to open a new dialog called "Secure media folder" with the media store tree structure. A folder from the media store can be selected that will be used to store the project's secure media. The "secure media folder" has a special designation in SiteArchitect . Use the delete icon to remove a selection that was already entered (see Chapter 11.3 page 538).



The settings defined here for "secure media" only affect the generation of the FirstSpirit preview. It is only possible to protect generated or deployed content from unauthorized access using the FirstSpirit security module³² (see Chapter 11.3 page 538).

³² Refer to the module documentation



Prefix for the access control database: prefix for completing the entry of a file's ACL database information. The full path to a file in FirstSpirit always consists of three parts:

- URL of web application (e.g. `http://myServer.com`)
- prefix for the access control database (`/fs5_security`)
- path to a file (`/de/index.html`)

The full path within the ACL database consists of the prefix and the path to the file, e.g. `/fs5_security/de/index.html`

In Figure 7-107 the prefix is `"fs5_security"` and is the equivalent of the direct subdirectory `"fs5_security"` of a web application such as `"live"`:
`"~Webserver/webapps/live/fs5_security"`.

The absolute prefix is required for the web application. The prefix is also in this case the last part of the `"Path on live server"` field value in the Deployment Servlet dialog (see Chapter 7.5.10.6.3 page 419).

For more information, see the documentation for the FirstSpirit security module.

Permission variable: the access permissions to objects can be defined through metadata in the media store. An input component is defined for this in the page template (`CMS_INPUT_PERMISSION`). Enter the name assigned to this component here. FirstSpirit will then decide whether a user can access an object or not using the input component value on the Metadata tab.

For more information, see the documentation for the FirstSpirit security module.

Allow access if no permissions are defined: if the checkbox is selected, access to `"secure media"` is permitted to all group members (from the `groups.xml` file) if no definition to the contrary has been defined using the permission variable. If there is no check mark in the box, only the values set using the permission variable will be evaluated.

*For more information on user permissions, see Chapter 11 page 528.
For more information on the "secure media" concept, see Chapter 11.3 page 506 ff.*



7.4.17 Project components

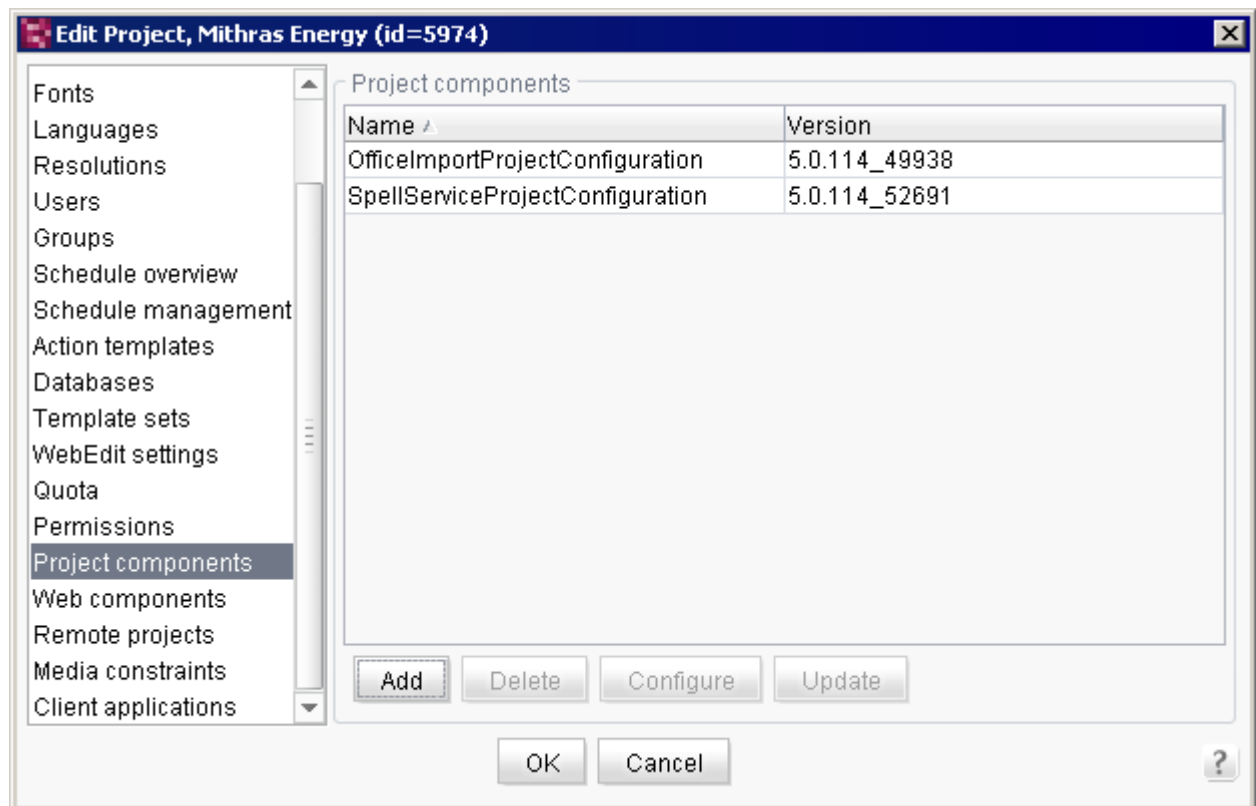


Figure 7-108: Project properties – Project components

Components are saved in this area which are to be available to the entire project (not only for individual areas of the project; see Chapter 7.4.18). Certain steps are required to install a project application. For instance, to install the portal component in a project, the corresponding module must be installed on the server (see 7.3.11 page 265). An example of a project component is the FirstSpirit portal component, which is part of the "FS Portal" module (see the FirstSpirit Portal documentation).

The "Add" button is then used to add the ("project application" type) components to a project. Once this is done, all functions of the installed component are available in the project. (In the case of portal components, portal folders within the project template store are displayed, for instance.)



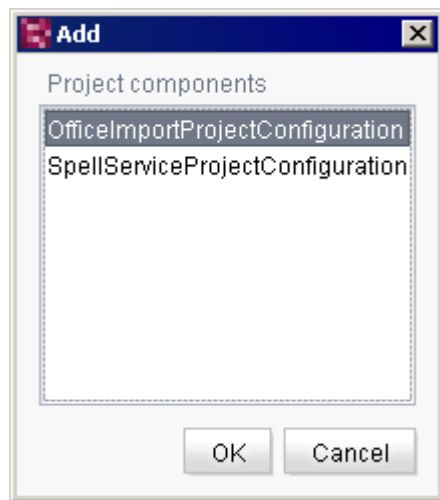


Figure 7-109: Adding a project component

The "Delete" button is used to remove ("project application" type) components that have already been added to a project.



When deleting a project component, all content associated with it as well as the component configuration are removed from the project.

The "Configure" button is used to edit a project component that had been previously added (see Chapter 7.3.11 page 265). Depending on the component, configuration takes place either via one of the GUIs generated by the component or a generic UI.

The "Update" button is used to update the component. The update compares the current version of the component to the component on the First Spirit server and then updates the one on the FirstSpirit server. If a newer version is available there, updating can be initiated for the particular project. Project component updates may result in adjustments within the project (e.g. configuration adjustments).



7.4.18 Web components

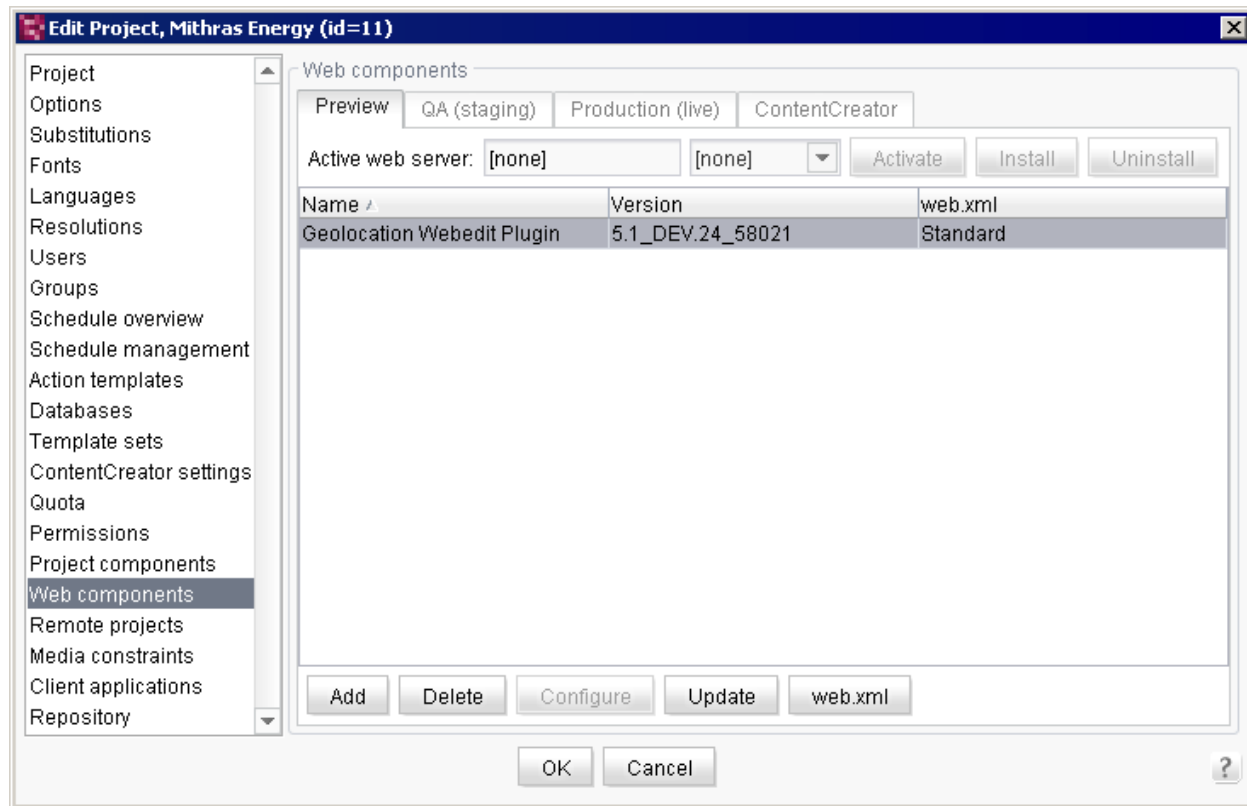


Figure 7-110: Project properties – Web components

Web components can be activated for a project in this area.

Web areas:



Figure 7-111: Web areas within a project

There are four different areas for each project: The web components for each area can be activated and configured individually on the respective tab:

- Preview: location where the project contents will be previewed.
- QA (staging): location for the generated project contents.
- Production (live): location for the deployed project contents.
- ContentCreator: configuration for a local project ContentCreator instance (see Chapter 5.2.2 page 191).



Web server area:

In every web area, web components can be configured for each project. The current **active web server** for each area is displayed in the dialog. The "internalJetty" entry is included by default. Additional web servers can be added as well. All web servers configured in the "Web server" area are available to choose from (see Chapter 7.3.12 page 271). The adjacent combo box is used to select a different web server. Different conversion steps are required depending on the type of web server:

- Internal web server (see Chapter 7.4.18.1 page 345)
- Generic web server (see Chapter 7.4.18.2 page 345)
- External web server (see Chapter 7.4.18.3 page 346)
- Tomcat (see Chapter 7.4.18.4 page 346)

The "Install" button is used to combine all web components in the particular web area of the project and to install them based on the configured web server. The button is activated if the web component has not yet been installed but is ready for installation. If the button is disabled, the web component has already been installed.

If the selected web server is an external or generic web server (without the required script functionality), the **"Download" button** is displayed instead (see Chapter 7.4.18.3).

If the web application has already been installed, but then the configuration has changed (see Chapter 7.4.18.7) or a component has been added (see Chapter 7.4.18.5) or deleted (see Chapter 7.4.18.6), the **"Update" button** is displayed instead of the "Install" button.

The "Uninstall" button is used to remove web components from the particular web area of the project. This action is performed for all web components of the particular area. Depending on the web server used, web component removal is carried out in a similar way as installation. If the button is disabled, the web component has not been installed yet.

The "Download" button is used to download an application's WAR file, which needs to be installed manually on the web server (see Chapter 7.3.13.5 page 282). The button is only displayed for configuring external web servers or generic web servers (without the required script functionality).

The "Activate" button switches the configuration of the project-specific web area to the selected web server, which is then displayed as the active web server for this area.



7.4.18.1 Configuring an internal web server for a web application

Control for the internal Jetty web server is provided by default and cannot be changed.

If a different web server was activated for a project-specific web area, the configuration can be reset to the internal web server by taking the following steps:

1. Select "internalJetty" from the combo box.
2. The "Install" button is active. Clicking on this button unpacks the WAR file to the target directory of the (Jetty) web server; the web components are then registered directly in Jetty.
3. The "Activate" button will be active after installation. Clicking on the button switches the configuration of the web area to the internal web server, which is then displayed as the active web server for the area.
4. All changes to the configuration must be confirmed and saved by clicking on "OK".

7.4.18.2 Configuring a generic web server for a web application

It is only possible to select a generic server if a corresponding web server instance was previously added to the server (see Chapter 7.3.12.1 page 273). Control of the generic web server is not provided by default, but it is possible using scripts (see Chapter 7.3.12.2 page 273). If these scripts are not available, the procedure is identical to that of the external web server (see Chapter 7.4.18.3).

If a generic web server is to be activated for a project-specific web area, the following steps are required:

1. Select the entry from the combo box for the desired generic web server.
2. The "Install" button becomes active. If the corresponding functionality has been provided via a script, clicking on the button will copy the WAR file, unpack it automatically to the respective web server and register the individual components on the web server (the relevant script-based web server control must be configured beforehand within "Server properties" (see Chapter 7.3.15.2 page 154)).
3. The "Activate" button will be active after installation. Clicking on the button switches the configuration of the web area to the generic web server, which is then displayed as the active web server for this area.
4. All changes to the configuration must be confirmed and saved by clicking on "OK".



7.4.18.3 Configuring an external web server for a web application

It is only possible to select an external server if a corresponding web server instance was previously added to the server (see Chapter 7.3.12.3 page 275). Control of an external web server is not supported via FirstSpirit and must be done manually (see Chapter 7.3.12.4 page 276). In the case of external web servers, only downloading of the WAR file is offered. Installation on the web server and registering the web components must be performed manually.

If an external web server is to be activated for a project-specific web area, the following steps are required:

1. Select the entry from the combo box for the desired external web server.
2. The "Download" button becomes active. Use this button to download the application's WAR file.
3. The WAR file must be installed on the external web server. The installation is either done manually via the administrative interface of the external web server or automatically from the web server file system.
4. After installation, the web area configuration can be switched to the external web server by clicking on "Activate". The external web server is now displayed as the active web server for the application.
5. All changes to the configuration must be confirmed and saved by clicking on "OK".

7.4.18.4 Configuring a Tomcat web server for a web application

It is only possible to select a Tomcat web server if a corresponding web server instance was previously added to the server (see Chapter 7.3.12.5 page 276). Control of a Tomcat web server is supported by FirstSpirit and takes place automatically (see Chapter 7.3.12.6 page 277). Installation on the web server and registering the web components is handled by FirstSpirit.

If a Tomcat web server is to be activated for a project-specific web area, the following steps are required:

1. Select the entry from the combo box for the desired Tomcat web server.
2. The "Install" button becomes active. This button copies the WAR file, which is automatically unpacked by the web server before the web components are registered.
3. The "Activate" button will be active after installation. Clicking on this button switches the configuration of the web area to the Tomcat web server, which is then displayed as the active web server for this area.
4. All changes to the configuration must be confirmed and saved by clicking on "OK".





If the `fs-server.jar` file has been updated, the steps described previously must be repeated manually for all web applications in all projects for which a "Tomcat" type web server was selected. Updates are not automatic.

7.4.18.5 Adding a web component

Add: clicking this button opens the "Add" dialog. All web components installed on the server are displayed in the list (see Chapter 7.3.11 page 265).

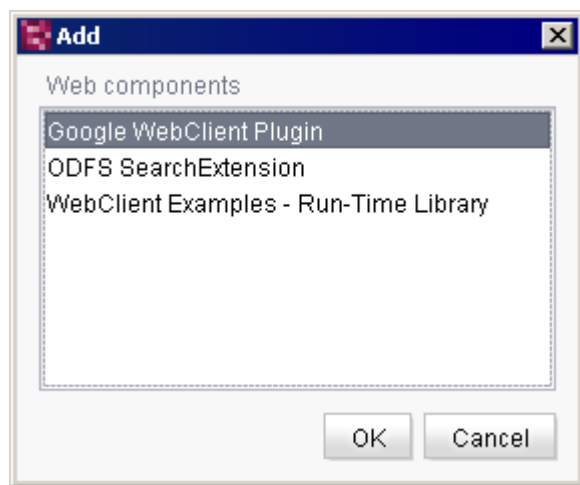


Figure 7-112: Adding a web component

These web components can be added to individual web areas (Preview, Staging, Live, ContentCreator) within the desired project. These components can then be configured either by using a GUI generated by the component or by using a generic GUI (see Chapter 7.4.18.7 page 348). The components still need to be activated after configuration. A component can be activated or deactivated within a project only for specific areas (see "Installation").

7.4.18.6 Deleting a web component

Use the "Delete" 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 when next deployed. For the changes to take effect, deployment on the server is required (see "Installation").



7.4.18.7 Configuring a web component

The "Configure" button is used to edit a component that had been previously added (see Chapter 7.3.11 page 265). Depending on the component, configuration takes place either via one of the GUIs generated by the component or a generic UI. The configuration dialog for FirstSpirit DynamicPersonalization appears as follows:

Configure

☐ Manual configuration Configure

Configuration

Name	Priority
------	----------

Add Remove Edit

☐ Use dummy user

☐ Activate group 'Everyone'

Group 'Everyone'

LOG4J default configuration file

SSO Cookie Domain

SSO Cookie Time-to-Live

SSO cookie name

OK Cancel

Figure 7-113: Configuring a local web component (example)

7.4.18.8 Updating a web component

Update: the "Update" button is used to update the component.. Updating compares the current version of the component to the one on the FirstSpirit server. If a newer version is available there, updating is initiated for the particular project. Web component updates for a project may result in adjustments within the project. This may be necessary to adjust the project



configuration, for instance.

7.4.18.9 Editing web.xml

If one or more web components are configured for a web area, a web.xml file is created automatically that consists of the individual web.xml files of the respective components. The web.xml file can be edited manually. Clicking on the "web.xml" button opens the dialog box where the file can be configured manually. After saving the changes, the "edited" value appears in the "web.xml" column of the overview instead of the "default" value.

The screenshot shows a 'Web components' dialog box with tabs for 'Preview', 'QA (staging)', 'Production (live)', and 'ContentCreator'. Below the tabs, there are input fields for 'Active web server' (both set to '[none]') and buttons for 'Activate', 'Install', and 'Uninstall'. A table lists the components:

Name	Version	web.xml
FIRSTpersonalisation	5.1_DEV.26_59132	Edited

At the bottom of the dialog, there are buttons for 'Add', 'Delete', 'Configure', 'Update', and a 'web.xml' button.

Figure 7-114: After manually editing the web.xml

7.4.19 Remote projects

This area contains a list of all remote projects configured for the project. Remote projects are projects from which media or data can be referenced or even loaded.

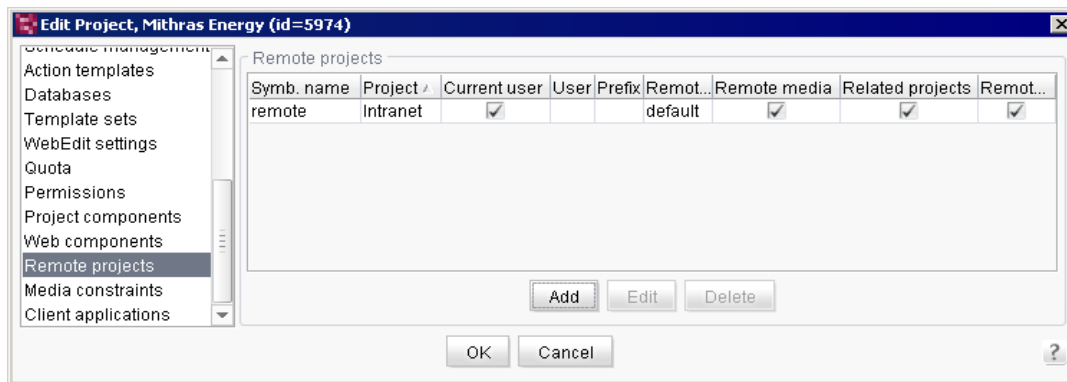


Figure 7-115: Project properties – Remote projects

The **Add** button opens a window where a new remote project can be configured. If a remote project already exists, the **Edit** button opens the same window.

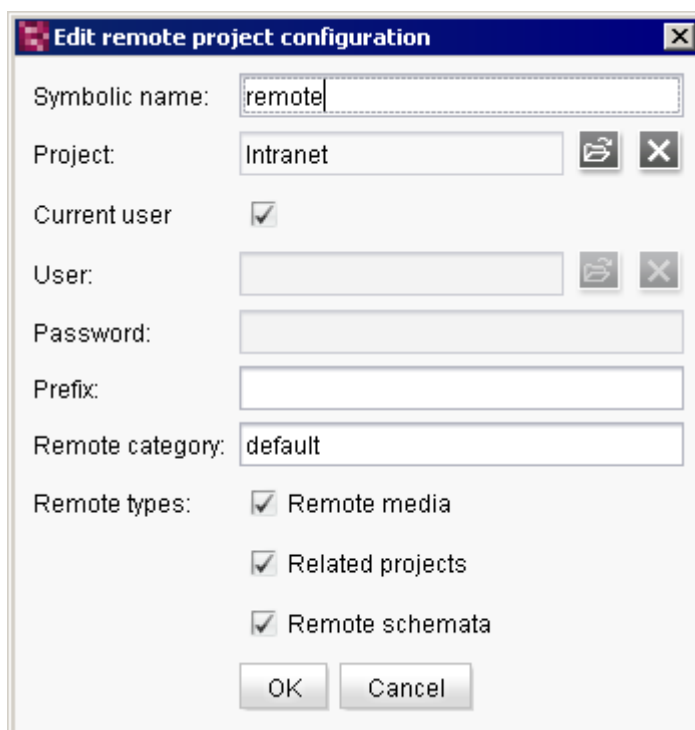


Figure 7-116: Configuring a remote project



Symbolic name: a unique name is specified here for the remote project. This name is used to reference the desired remote project in the target project.



Only projects that are on a server with the target project can be configured as remote projects.

Project: the file symbol is used to select the desired project (from which media/data are to be referenced) from the server project selection list.

Current user: if this option is selected, the system checks if the current user has the required project permissions if an attempt is made to access the remote project.

User: the file symbol is used to select a user from the user list of the server that controls the user's permissions for access to the remote project.

Password: the selected user's password must be entered in this field.



The technical user must be known as a user in the target project and must have at least permission to view the entire media store or individual folders within the remote project media store.

Prefix: the name of the remote project must be entered again in this field. The prefix is required for generating the URL when generating the target project.

Remote category: a category designation for remote project configuration is specified in this field. This can be used to combine multiple remote projects into one group. The category can be used in a reference configuration in order to be able to select from the defined group of remote projects. By default, the field is pre-populated with the value "default" (default category).

Remote types: three different types of remote access are possible:

- Remote media (see Chapter 7.4.19.1 page 352)
- Related projects (see Chapter 7.4.19.2 page 352)
- Remote schemata (see Chapter 7.4.19.3 page 353)





All of the remote access types involve additional license-dependent functions.

7.4.19.1 Remote media concept

The objective of the "remote media" concept is to create all media in a separate media project and to manage the files from this central location. All FirstSpirit projects affected can then access the media inventory (images and files) using remote media access.

Unlike distributing media using the license-dependent "CorporateContent" function, the media do not have to be imported to the involved projects, but instead they can be referenced directly through remote media access. The objects physically remain in the media project, but they can be used in any desired project.

If a valid license exists for the functionality, remote media access can be activated via FirstSpirit ServerManager.

For more information on configuring and using remote media, see the "FirstSpirit Corporate Media" functionality documentation, the documentation about FirstSpirit SiteArchitect and FirstSpirit Online Documentation.

7.4.19.2 Related projects concept

The "related projects" concept involves links from one project to another FirstSpirit project. These links can be made indirectly by defining link targets in a FirstSpirit project (via specifically configured input components), but also by directly defining the targets within the site store.

If a valid license exists for the functionality, "related projects" access can be activated via FirstSpirit ServerManager.

For more information on configuring and using related projects, see FirstSpirit Online Documentation.



7.4.19.3 Remote schemata concept

The "remote schemata" concept is required for the license-dependent "FirstSpirit DynamicDatabaseAccess" module, which is used to bind different database technologies to FirstSpirit. Using the module, content from a database can be viewed and edited via a web application. FirstSpirit DynamicDatabaseAccess uses database schemata from the FirstSpirit template manager. These schemata can be re-defined within a project. A graphical editor for editing a database schema is available for creating the desired database schema. Each schema can access existing database structures or create new table structures in an existing database (for more information on using database schemata, see *FirstSpirit Online Documentation*).

Remote access to database schemata from other FirstSpirit projects is also possible if the "remote schemata" remote type has been activated in the remote project configuration.

For more information on FirstSpirit DynamicDatabaseAccess, see the corresponding module documentation.



7.4.20 Media constraints

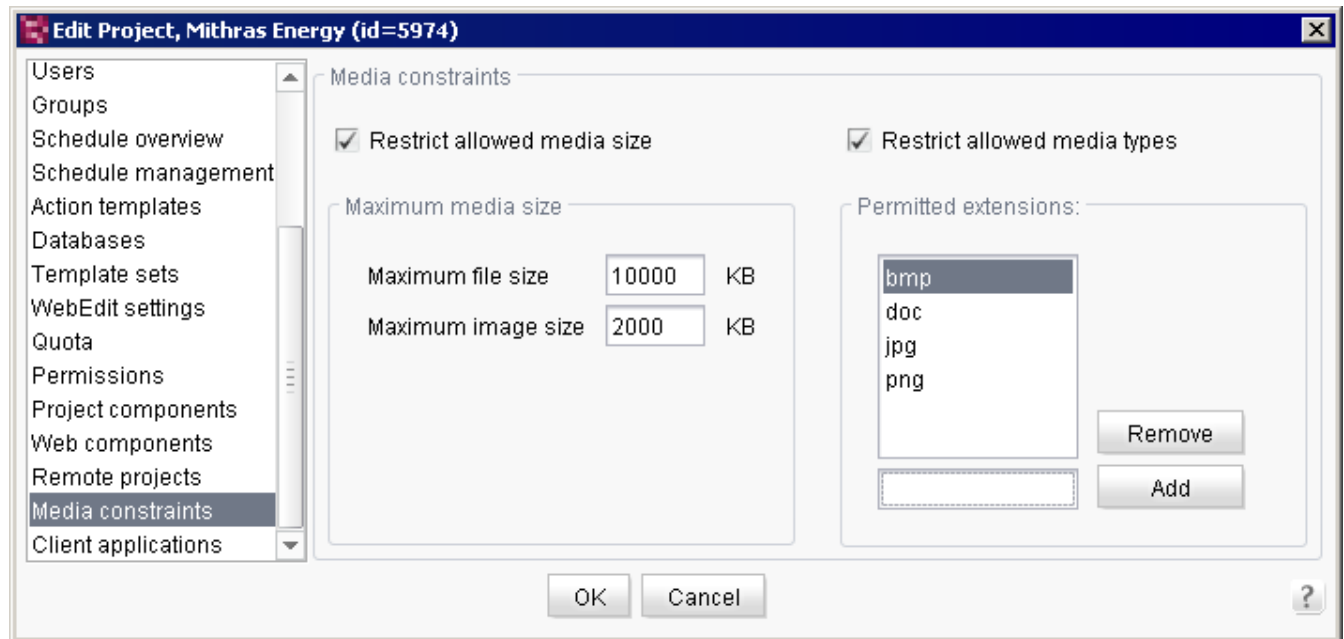


Figure 7-117: Project properties – Media constraints

The media uploaded to the media store can be limited to certain file sizes and/or formats using the options "Restrict allowed media size" and "Restrict allowed media types". Both options can be activated separately or combined.



The constraints are used only when uploading new media; subsequent constraints will not affect existing media.

7.4.20.1 Restrict allowed media size

Restrict allowed media size: if this checkbox is selected, the input fields for entering a maximum file or image size are active.

Maximum file size: the maximum file size (in KB) for "file" type media uploads to the media store can be defined in this field. Files that are larger than the specified size limit cannot be added.

Maximum image size: the maximum file size (in KB) for "image" type media uploads to the media store can be defined in this field. Images that are larger than the specified size limit can no



longer be added.

Only integers are permitted in these two fields. Entering other values, such as letters or characters, is immediately prevented, and the particular field starts flashing red.

7.4.20.2 Restrict allowed media types

On the right half of the media constraints section under the **Restrict allowed media types** selection box of the configuration interface is an option for limiting the file format for uploadable media. This is done by entering the file extensions.

Permitted extensions: the permitted file extensions can be added or removed here. Checking the file format is done only based on the file extension and not by analyzing the file contents. All extensions must be specified here without the ".". Entry is case sensitive, which means that subsequently only files with the .DOC extension and not those with the .doc extension are prevented from being uploaded if the restriction is applied to the DOC media type. In case of doubt, different spellings should be entered in order to cover all files of the desired format.

The **"Add"** button is used to add and save an extension previously defined in the input field to the left of the "Add" button for the list of permitted extensions. The extensions are listed in alphabetical order.

The **"Remove"** button removes a file extension that has already been selected from the list.

7.4.20.3 Displaying the constraints in SiteArchitect

The media constraints are displayed to the editor in the form of a filter constraint in the SiteArchitect file selection dialog, if this is activated in the project configuration. This also applies to the media wizard. Media that are larger than the defined size limit and/or have a file extension that is not permitted are not available as a selection in the file selection dialog.



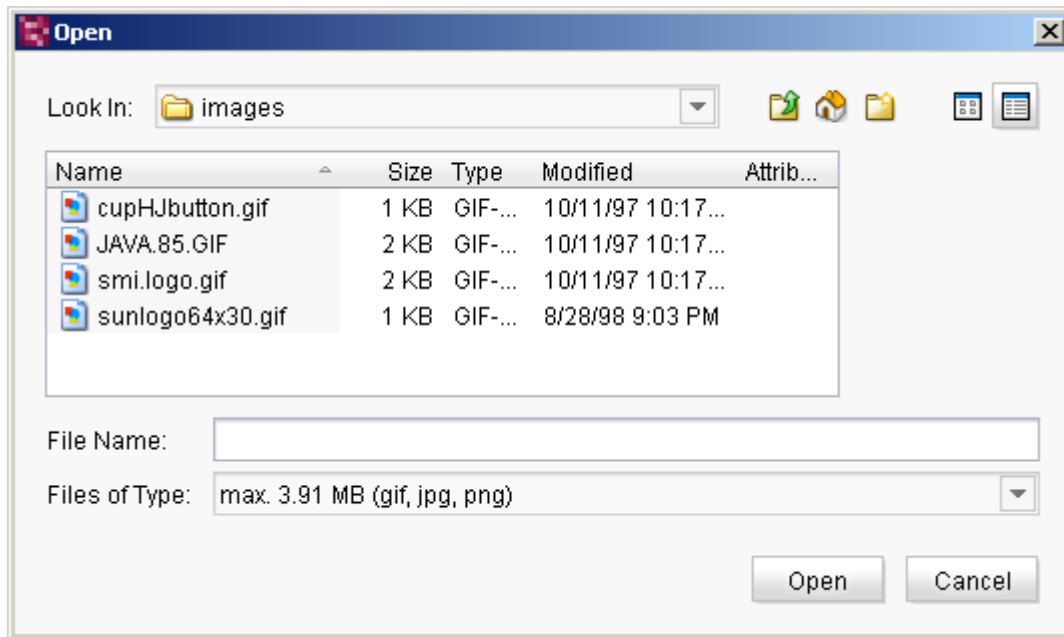


Figure 7-118: Activated file size and file extension restrictions

The restrictions are displayed once more to the right of "Files of type:". In the example in Figure 7-118, the files with the file extensions "jpg", "png" and "gif" up to a size of 1.95 MB may be selected and uploaded.



If the files that are to be copied via drag-and-drop from the directory structure of the workstation and then pasted to the media store are not permitted due to the media constraints, a relevant warning will appear. The files will not be added to the media store.

7.4.20.4 Activated restrictions in ContentCreator

The same technical implementation of filter rules in ContentCreator is not possible in SiteArchitect. Since the file selection dialog available in the browser for uploading media is not an integral FirstSpirit implementation, but is instead a firmly integrated part of every browser (e.g. Firefox, Mozilla, Internet Explorer, Opera), filtering technically cannot be done the same way as in SiteArchitect. The files are therefore filtered only after uploading and, if necessary, an error message appears to the user if the files exceed the media constraints defined in the project configuration.



7.4.21 Client applications

The settings made here affect the "View" menu in SiteArchitect. To ensure the changes take effect, the affected project must be restarted.

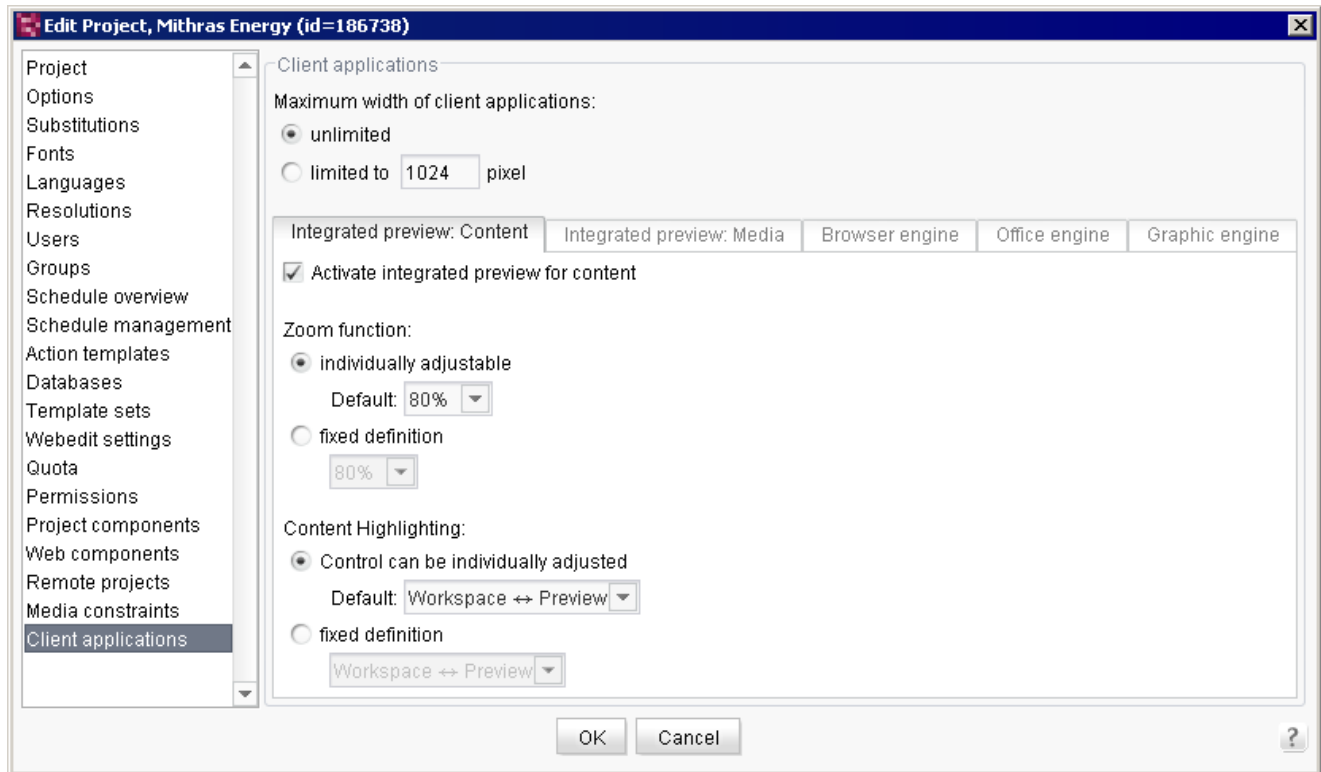


Figure 7-119: Project properties – Client applications

Maximum width of client applications: here you can specify whether the width should be limited to a certain number of pixels or whether it should be customizable by the user.

7.4.21.1 Integrated preview: Content

Activate integrated preview for content: this option depends on the setting for the "Activate browser engine" option on the "Browser engine" tab. The two options must be either enabled or disabled at the same time. If this option is not checked, changes cannot be made to this tab.

If this option is selected, the integrated preview for content for the selected project can be used in SiteArchitect, which means that editors can select individually whether they want to work with the integrated preview or not using the menu function "View" / "Integrated preview – use for content" (see *FirstSpirit Manual for Editors (SiteArchitect)*). If the box is disabled, the menu function "View" / "Integrated preview – use for content" is not available as an option in SiteArchitect. This



option is selected by default.

Zoom function

individually adjustable: if this option is selected, the users themselves can select the scaling level in the particular project. The "Default" combo box allows the user to specify which scaling level should be preselected. The default setting is 80%.

fixed definition: if this option is selected, the selected scaling level is used project wide. The scaling cannot be changed by editors.

Content Highlighting

Control can be individually adjusted: if this option is selected, editors can individually adjust the Content Highlighting behavior using the menu item "View" / "Content Highlighting Control" (see *FirstSpirit Manual for Editors (SiteArchitect)*). The "Default" combo box allows the user to specify what the predefined behavior should be. The default setting is "Workspace ↔ Preview"

fixed definition: if this option is selected, the selected behavior is used project wide. The editors cannot change the behavior.



7.4.21.2 Integrated preview: Media

The user can specify here whether the integrated preview for media can basically be used in the particular project and which particular file formats are to be used with the various applications.

Integrated preview: Content **Integrated preview: Media** Browser engine Office engine Graphic engine

☒ Activate integrated preview for media

Use office engine for the following file extensions (comma-separated):

odt,ott,sxw,doc,docx,odg,otg,sxd,ods,ots,sxc,cls,xls,xlsx,odf,sxm,odp,otp,sxi,ppt,pptx,odm,oth,odb

Use browser engine for the following file extensions (comma-separated):

pdf,html,htm,url,swf

Use integrated text editor for the following file extensions (comma-separated):

css,js,txt,xml,csv,json,as

Use integrated image view for following file extensions (comma-separated):

png,jpg,jpeg,bmp,gif,psd

Use Microsoft Windows Media Player (Windows only) for the following file extensions (comma-separated):

avi,mpg,mpeg,wmv,asf,mp3,mp4

Restore default values

Figure 7-120: Default settings for Integrated preview: Media

Activate integrated preview for media: this option depends on the setting for the "Activate office engine" option on the "Office engine" tab and "Activate graphic engine" option on the "Graphic engine" tab. If one or the other option is activated, this option will also be activated. If this option is disabled, changes cannot be made to this tab.

If this option is selected, the integrated preview for media for the selected project can be used in SiteArchitect, which means that editors can select individually whether they want to work with the integrated preview or not using the menu function "View" / "Integrated preview – use for media" (see *FirstSpirit Manual for Editors (SiteArchitect)*). If the box is not checked, the menu function "View" / "Integrated preview – use for media" is not available as an option in SiteArchitect.

The option is selected by default in new projects, making it possible to assign file formats to applications in the integrated preview for display using the following fields.



Use office engine for the following file extensions (comma separated): file extensions for files that are to be displayed using the office engine (Microsoft Office, OpenOffice or Google Docs) can be entered in this field with commas to separate them. The file formats included here by default are for word processing, spreadsheet and presentation programs.

Use browser engine for the following file extensions (comma separated): file extensions for files that are to be displayed using the browser engine (Microsoft Internet Explorer and Mozilla Firefox) can be entered in this field with commas to separate them. The file formats included here by default are for files that can be displayed by web browsers or browser plug-ins.

Use internal text editor for the following file extensions (comma separated): file extensions for files that are to be displayed using the FirstSpirit internal text editor can be entered in this field with commas to separate them. The file formats included here by default are for files that can be created and edited using text editors.

Use internal picture viewer for the following file extensions (comma separated): file extensions for files that are to be displayed using the graphic engine (simple image editing, Advanced Image Editor, Picnik, Pixlr) can be entered in this field with commas to separate them. The file formats included here by default are image file formats.

Use Microsoft Windows Media Player (only Windows) for the following file extensions (comma separated): file extensions for files that are to be played using Windows Media Player can be entered in this field with commas to separate them. The file formats specified here by default are audio and video file formats. Windows Media Player can only be used in conjunction with Microsoft Windows. This field is grayed out if a different operating system is in use.

Restore default values: clicking on this button will restore the default settings.



7.4.21.3 Browser engine

The screenshot shows the 'Browser engine' tab of a configuration window. At the top, there are five tabs: 'Integrated preview: Content', 'Integrated preview: Media', 'Browser engine' (selected), 'Office engine', and 'Graphic engine'. Below the tabs, the 'Activate browser engine' checkbox is checked. There are two radio button options: 'individually adjustable' (selected) and 'fixed definition'. Under 'individually adjustable', there is a 'Default:' label followed by a dropdown menu showing 'Mozilla Firefox (v15)'. Under 'fixed definition', there is a dropdown menu also showing 'Mozilla Firefox (v15)'.

Figure 7-121: Project properties –Integrated preview: Browser

Activate browser engine: this option depends on the setting for the "Activate integrated preview for content" option on the "Integrated preview: Content" tab. The two options must be either enabled or disabled at the same time. If this option is disabled, changes cannot be made on this tab.

The file formats defined in the "browser engine" field of the "Integrated preview: Media" tab (see Chapter 7.4.21.2 page 359) are then not displayed in the browser engine integrated in FirstSpirit, but are instead displayed in an external application. This option is selected by default so that the user can specify which browser engine is to be used for the file formats defined in the "browser engine" field using the following radio buttons and combo boxes:

individually adjustable: if this option is selected, users can select the browser engine in the particular project themselves. This option is selected by default. The "Default" combo box allows the user to specify which browser engine should be set by default. The default setting is Mozilla Firefox.

fixed definition: if this option is selected, a browser engine can be specified for the project. It will then not be possible for the user to make a selection from the "Browser engine" submenu of the SiteArchitect "View" menu in this project.

In FirstSpirit version 5.1.R2 Google Chrome has been integrated. Known restrictions:

- The integration of Chrome is based on a special application and does not use any existing, locally installed version of Google Chrome, or any user data that is used for this purpose. No automatic updates are carried out either.
- No plug-ins can be installed (e.g., Adobe PDF plug-in for displaying PDFs, Adobe Flash Player plug-in for displaying Flash files). This also means, for example, that no help PDF files can be displayed either ("Help"/"Users (SiteArchitect)", "Help"/"Users (ContentCreator)", and "Help"/"Administrators" menu items).





The integration of Google Chrome is currently in the BETA test phase and has not yet been officially released!

For information on the system requirements and limitations related to the browser engine, see current *FirstSpirit Technical Data Sheet*, and for information on the rollout process for native browser integration components, see Chapter 4.10 page 184.

7.4.21.4 Office engine

Integrated preview: Content Integrated preview: Media Browser engine **Office engine** Graphic engine

☒ Activate office engine

☒ individually adjustable

Default:

☐ fixed definition

Figure 7-122: Project properties – Default settings for Integrated preview: Office



A valid license is required for this function: the value for the `license.OFFICE_INTEGRATION` parameter in the license file `fs-license.conf` must be set to 1. Otherwise, the "Office engine" tab will be grayed out, it will not be possible to adjust any settings and office documents will not be viewable in the SiteArchitect integrated preview.

Activate office engine: this option depends on the setting of the "Activate integrated preview for media" option on the "Integrated preview: Media" tab (see Chapter 7.4.21.2 page 359). If the "Activate office engine" option is selected, the "Activate integrated preview for media" option will also be selected automatically.

If the "Activate office engine" option is disabled, changes cannot be made on this tab. The file formats defined in the "office engine" field of the "Integrated preview: Media" tab (see Chapter 7.4.21.2 page 359) are then not displayed in the integrated preview for media, but are instead displayed in an external application.

If a valid license exists, this option is selected by default so that the user can specify which application is to be used for the file formats defined in the "office engine" field using the following



radio buttons and combo boxes:

individually adjustable: if this option is selected, users themselves can select the office engine in the particular project. The "Default:" combo box allows the user to specify which office engine should be preselected. The default setting is Microsoft Office. Microsoft Office cannot be used for non-Windows operating systems. Therefore, "OpenOffice" (currently available only for beta testing) should be selected here.

fixed definition: if this option is selected, an office engine can be specified for the project. It will then not be possible for the user to make a selection from the "Office engine" submenu of the SiteArchitect "View" menu in this project. This option is selected by default in new installations.

Note on using OpenOffice: in order to be able to use OpenOffice, OpenOffice must be installed directly in the default program directories of the particular operating system (not in a subdirectory). For instance, in Microsoft Windows it would be installed under "C:\Program Files" or "C:\Program Files (x86)", and on Linux it would be under "/usr/lib", "/usr/local/lib" or "/opt". Otherwise, OpenOffice documents might open in a separate window instead of in the integrated preview.



To use applications in the integrated preview, it must be noted that FirstSpirit provides the necessary interfaces for integrating the application, but usually does not have any control over the integrated applications. Integrated external applications are not included as part of the FirstSpirit. This means, among other things, that the manufacturer, customer or partner who developed the integrated application is responsible for its functionality.



*The simultaneous use of the integrated preview for media with Microsoft Office and as an external Microsoft Office desktop application (e.g. launched from Windows or from SiteArchitect) may cause problems under certain circumstances and is not guaranteed to work when integrated with FirstSpirit. In this case, either the integrated preview for office documents **or** the Microsoft Office desktop application should be used in FirstSpirit, but **not** both.*

Notes on using Google Docs: Google Docs can also be used to edit office documents in FirstSpirit SiteArchitect on the Mac OS X. To do this, an active connection to the Internet and a Google account are required (<http://docs.google.com>). It is no longer necessary to install the software locally in order to create and edit office documents, but all documents must be uploaded to a Google server before being edited. Google Docs integration is not guaranteed to work and is



not authorized for use in production due to the sensitive nature of data protection, the still very strict limitations on document editing (e.g. file size limitations and incompatibilities) and the use of an unapproved API. In some cases documents could actually become damaged if they are edited in Google Docs. This problem does not stem from FirstSpirit.

The free version of Google Docs supports only a maximum file size of 1 MB. Currently, Google Docs can only be used with the "Internet Explorer" browser engine in SiteArchitect.

7.4.21.5 Graphic engine



Integrated preview: Content Integrated preview: Media Browser engine Office engine **Graphic engine**

☒ Activate graphic engine


☒ individually adjustable

Default:

☐ fixed definition

Figure 7-123: Project properties – Default settings for Integrated preview: Graphic

Activate graphic engine: this option depends on the setting of the "Activate integrated preview for media" option on the "Integrated preview: Media" tab (see Chapter 7.4.21.2 page 357). If the "Activate graphic engine" option is selected, the "Activate integrated preview for media" option will also be selected automatically.

If the "Activate graphic engine" option is disabled, changes cannot be made to this tab. The "Graphic engine" entry in SiteArchitect in the "View" menu is grayed out and disabled. Pictures can still be edited using the familiar functions ( icon).

If the option is selected, the following radio buttons and combo boxes can be used to specify which applications are to be used:

individually adjustable: if this option is selected, users themselves can select the graphic engine in the particular project. The "Default:" combo box allows the user to specify which graphic engine should be preselected.

fixed definition: if this option is selected, a graphic engine can be specified for the project. It will then not be possible for the user to make a selection from the "Graphic engine" submenu of the SiteArchitect "View" menu in this project.





To use applications in the integrated preview, it must be noted that FirstSpirit provides the necessary interfaces for integrating the application, but usually does not have any control over the integrated applications. Integrated external applications are not included as part of the FirstSpirit. This means, among other things, that the manufacturer, customer or partner who developed the integrated application is responsible for its functionality.



*When using the **Advanced Image Editor**, the release is explicitly "not guaranteed to work" due to the application itself. This means that e-Spirit does not offer any explicit or implicit guarantee that the image editor functions will work. Use of the release is on an "as is" basis. If using image editing functions is critical for production, an external editing software with the appropriate manufacturer support should be used. Use of the Advanced Image Editor on the Mac OS is currently severely limited.*



*An active Internet connection is required in order to use the **PicMonkey** or **Pixlr** options. The images to be edited are effectively uploaded to and edited on the server of the respective provider. Data protection and privacy should be taken into consideration when using the editors.*



7.4.22 Repository

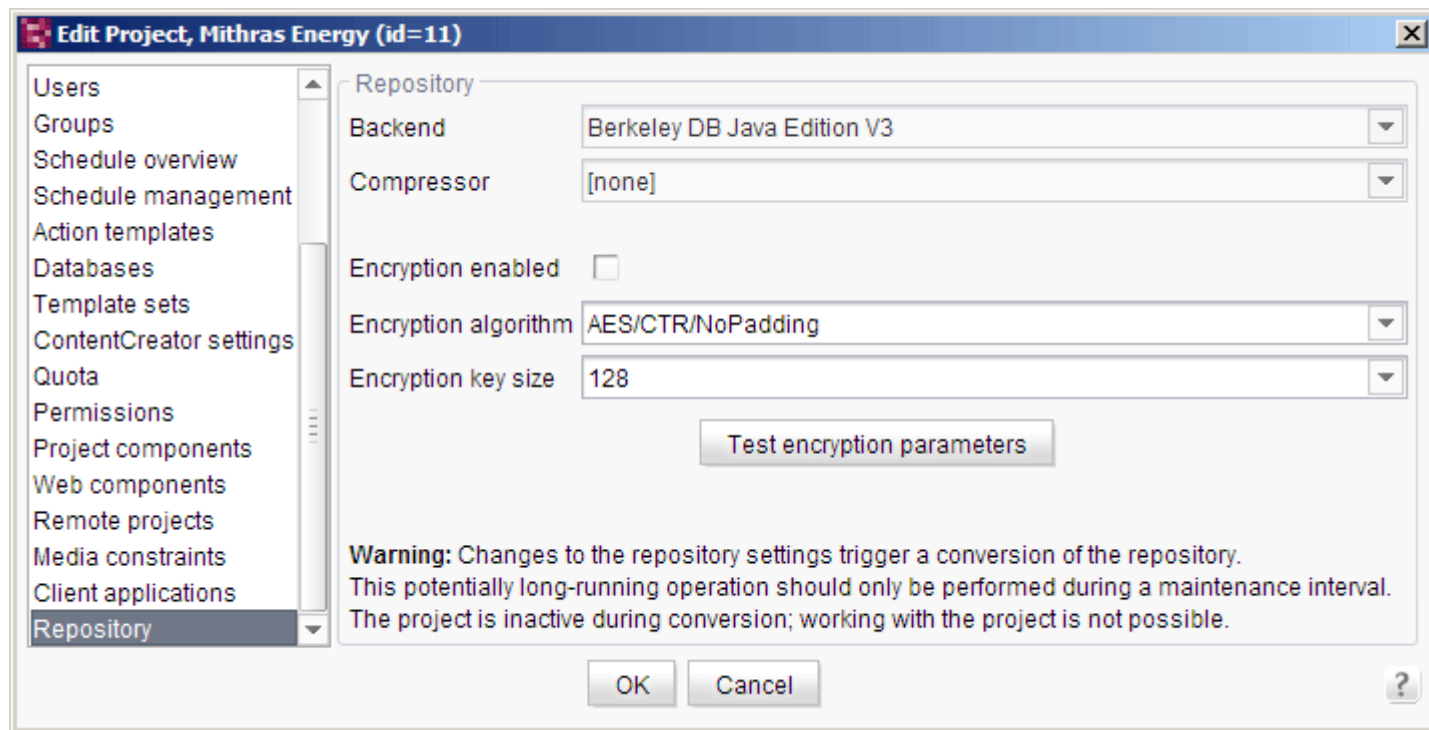


Figure 7-124: Project properties – Repository

In addition to continuing to offer the reliable 3.x version of Oracle Berkeley DB as standard as a repository for saving content data, FirstSpirit version 5.1 will also include version 5.x in order to provide the latest performance advantages over the previous version in terms of technical development and to stay on top of future development trends. For reasons of compatibility, both versions can be used in tandem on the same server within different projects. Migration from one version to the other is as easy as selecting it from a list.



The new Berkeley version 5.x was officially released following extensive quality assurance checks; however, as this is a significant version upgrade and it concerns a third-party product, this version should be tried out on a test system before being launched on production systems.

In order to operate different types of repository on a FirstSpirit Server (in different projects) at the same time, the repositories have been moved into modules:

- fs-berkeleydb3.fsm
- fs-berkeleydb5.fsm



This means that there is now an infrastructure available which can be used for other customer-specific repositories if required.

Backend: The required repository can be selected from this drop-down list.

Compressor: A different compression can be set as necessary via this drop-down list. Compression affects both the amount of disk space required and the access speed.

- [none]: No compression is used (recommended default setting).
- Deflate: Algorithm with high compression ratio
- Snappy: Algorithm developed by Google and designed to provide high speeds
- LZ4: Algorithm designed to provide high speeds

Repository encryption (FirstSpirit Version 5.1R4 and higher): This area can be used to configure encryption of the repository contents (content, structures, media) for a project. For background information on repository encryption, see Chapter 4.8.2 (page 153 ff).

Requirements: To configure these project settings, a global server key file is required (see Chapter 4.8.2.1, page 153). The `repository.encryption.keyFilePath` parameter must be used to store the path to the key file in the `fs-server.conf` configuration file (for information on configuration, see Chapter 4.3.1.11, page 56).

Note concerning default values: In the case of new and imported projects (with FirstSpirit Version 5.1R4 and higher), default values can be assigned to the form fields using global values from the `fs-server.conf` configuration file. This means that encryption can be enabled for a new project right away because the `repository.encryption` parameter is set to a value of 1 in the `fs-server.conf` configuration file (see Chapter 4.3.1.11, page 56). The project settings can be configured independently of the global values. As a result, encryption can be enabled or disabled for specific projects. In addition, specific projects can be encrypted with stronger or weaker algorithms as required.

Encryption enabled: This checkbox can be used to enable/disable repository encryption for each project on an individual basis. If this checkbox is *selected*, the encryption mechanism is applied to the current project (see also Chapter 4.8.2.2, page 154), but if it is *unchecked*, the repository contents are not encrypted. In the case of existing projects, the checkbox must be selected initially in order for pre-existing repository content to be encrypted. The project cannot be used during the initial encryption process (temporarily disabled). If encryption is disabled, the repository is decrypted using a similar process. A default value can be assigned globally by using the `repository.encryption` parameter in the `fs-server.conf` configuration file (for new



or imported projects in FirstSpirit Version 5.1R4 and higher) (see Chapter 4.3.1.11, page 56).

Encryption algorithm: The encryption algorithm to be used can be selected from the drop-down menu. The name, mode, and padding are specified in each case. The algorithms shown here are merely examples and are not to be construed as recommendations:

- AES/CBC/PKCS5Padding
- AES/CTR/NoPadding (**default**)
- ARCFOUR
- Blowfish/CBC/PKCS5Padding
- Blowfish/CTR/NoPadding

As an alternative to the suggested values, you can enter a valid encryption algorithm in the field manually. The actual encryption process is handled by the Java Cryptography Extension. As a result, it is the Java platform used that determines which symmetric encryptions and modes can be configured here. For details of which algorithms, modes and key sizes are possible, please see the relevant JCE documentation:

- <http://docs.oracle.com/javase/7/docs/technotes/guides/security/StandardNames.html#Cipher>
- <http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html#SunJCEProvider>

A default value can be assigned globally by using the `encryption.algorithm` parameter in the `fs-server.conf` configuration file (for new or imported projects in FirstSpirit Version 5.1R4 and higher) (see Chapter 4.3.1.11, page 56).

encryption.keySize: The length of the encryption key can be configured via this drop-down menu. As an alternative to the suggested values (64, 128, 192, and 256 bits), other values can be entered manually. The key size must be compatible with the configured algorithm (encryption, modes).

For more information on permitted key sizes, see:

- <http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html#SunJCEProvider>
- <http://docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders.html#importlimits>

A default value can be assigned globally by using the `encryption.keySize` parameter in the `fs-server.conf` configuration file (for new or imported projects in FirstSpirit Version 5.1R4 and higher) (see Chapter 4.3.1.11, page 56).



Test encryption parameters: Click this button to test the encryption process with the currently configured parameters. The test checks the encryption process in accordance with the specifications of the Java Cryptography Extension, e.g. to see whether the specified algorithm and key size can be combined with one another.

When the user confirms the selection with "OK", the system starts to convert the data using the desired settings. The relevant project is deactivated during the process.



To prevent any data loss, anyone using the project should log off first. Changes to the repository settings can take some time and should only be performed during a maintenance interval.



7.5 Schedule entry planning

FirstSpirit schedule entry planning can be used to group associated actions together into a schedule entry and to start them at defined, scheduled times.

The FirstSpirit ServerManager offers the ability to display existing schedule entries and the associated information. The "Schedule overview" table can be opened within the server or project properties (see Chapter 7.5.1).

The "Schedule management" menu item is used to add new schedule entries and manage existing ones (see Chapter 7.5.2). Creating new schedule entries is divided between configuring the schedule properties (see Chapter 7.5.4), such as entering a start time, and adding the desired actions that are to be carried out as part of the schedule entry (see Chapter 7.5.5).

The actions must first be created using ServerManager (see Chapter 7.5.6). Server-based actions are defined within the server properties (see Chapter 7.5.6.2); project-based actions are defined within the project properties of the individual projects (see Chapter 7.5.6.1).

The following server-based actions can be selected in the server properties:

- Clean up log files see Chapter 7.5.9.2 page 394
- Report analysis, compaction and transmission see Chapter 7.5.9.3 page 395
- Server update see Chapter 7.5.9.3 page 395
- Maintenance mode see Chapter 7.5.9.5 page 397

The following project-based actions can be selected in the project properties:

- Archive old project states see Chapter 7.5.10.1 page 402
- Content Transport (create, update, install)
 this is a schedule for automatizing Content Transport functionalities; for
 further information please see FirstSpirit "Corporate Content" module
 documentation
- Enterprise backup: Project data backup
 this is a license-dependent function;
 see the FirstSpirit "EnterpriseBackup" module documentation
- Execute generation see Chapter 7.5.10.2 page 406
- Execute project backup see Chapter 7.5.10.3 page 412
- Repair references see Chapter 7.5.10.4 page 412
- Rebuild search index see Chapter 7.5.10.5 page 412
- Execute deployment see Chapter 7.5.10.6 page 413



The two action types

- Send e-mail see Chapter 7.5.9.1 page 392
- Execute script see Chapter 7.5.9.4 page 395

can be used both in server-based and project-based schedule entries.

Actions already created once can easily be copied from the existing schedule entries (see Chapter 7.5.7). If the same actions are to be reused in multiple schedule entries, an "action template" can be created similarly to using the FirstSpirit template concept (see Chapter 7.5.3). The action template is used to add an action to any number of schedule entries (see Chapter 7.5.8). Changes to an action in this case are entered centrally through the template.

In addition to execution through ServerManager, schedule entries can also be executed interactively through the FirstSpirit SiteArchitect. To do this, the user needs to have the required permissions for interactive execution (see Chapter 7.5.4 page 380), which means the user must be in the list of authorized users or a member of a user group that is in the list of authorized groups.



7.5.1 Schedule overview

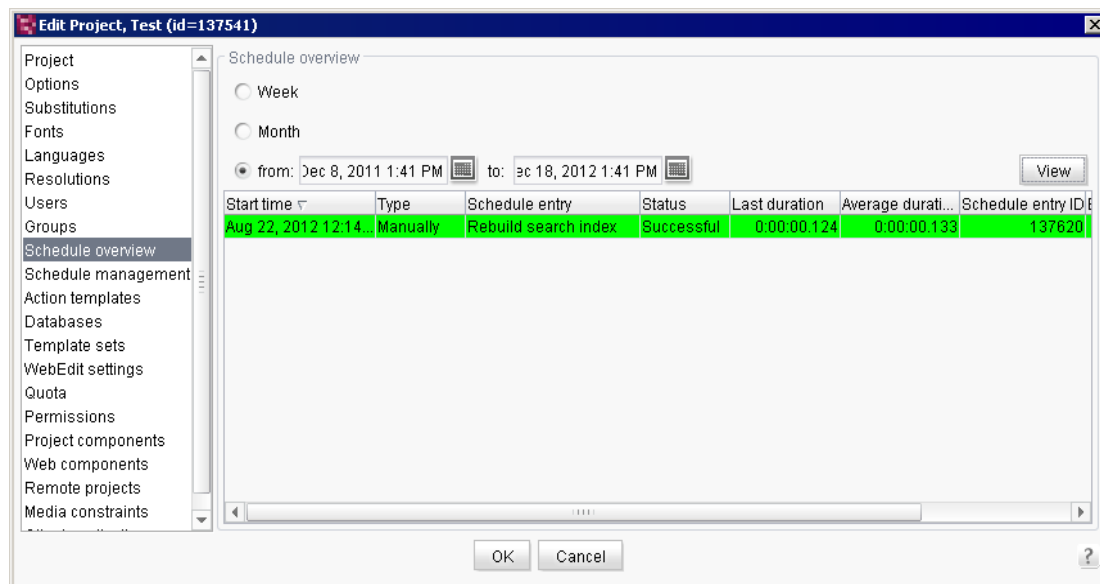


Figure 7-125: Schedule overview

The schedule overview table contains historic data up to the current date and calculated data for upcoming, active (not manual) schedule entries (see Chapter 7.5.1.2 page 373). Depending on whether the user is in the server properties (see Chapter 7.3.9) or the project properties (see Chapter 7.4.9), either only server-based or only project-based schedule entries will be displayed.

Schedule entries within a period of 5 days before the current date and up to 5 days after the current date are displayed by default. The user can choose to display all data from the current week, current month or from a defined period (see Chapter 7.5.1.1 page 372).

Double-clicking on the desired schedule entry opens the schedule details dialog (see Chapter 7.5.1.3 page 374).

7.5.1.1 Defining the overview period

Options – Week: if this option is *selected*, all schedule entries are listed that have been or are to be carried out during the current week (Monday through Sunday).

Options – Month: if this option is *selected*, all schedule entries are listed that have been or are to be carried out during the current month.



Options – from ... to: if this option is *selected*, all schedule entries are listed that have been or are to be carried out within the defined period. The start and end dates can each be selected by clicking on the calendar icon.

If a new option or a different start or end date has been selected, the selection must be confirmed by clicking on the "View" button. The table will then be updated with the requested data.

7.5.1.2 Schedule overview table

The table includes the following columns:

Start time: the time at which the schedule entry was initiated or will be initiated.

Type: there are different types of execution methods available for schedule entries (see page 380).

- Manually: schedule entry is manually executed
- Schedule entries that are executed automatically can be distinguished as follows:
 - Once
 - Periodically

Schedule entry: unique name of schedule entry. The name is defined when a new schedule entry is added.

Status: this column describes the execution status of the schedule entry. The following status types are available:

- Not started
- Running...
- Canceled
- Errors
- Completed with errors
- Successful

The rows of the table will appear in different colors depending on the status:

- Green: successful
- Red: errors
- Yellow: schedule conflict



Last duration: the period when the last schedule entry was executed is displayed here.

Average duration: the average duration, as well as the "last duration", is part of the statistical information that can be useful for scheduling future times.

7.5.1.3 Details of schedule entry

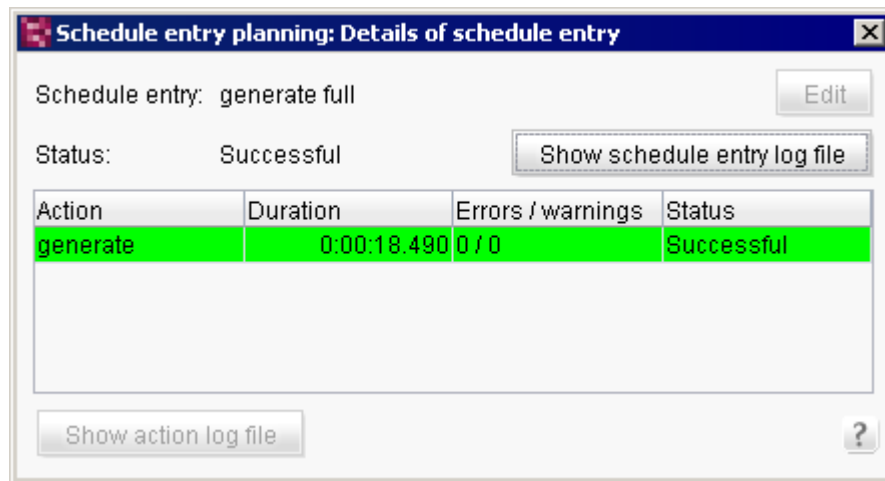


Figure 7-126: Details of schedule entry

Double-clicking on the schedule entry from the schedule overview (see Figure 7-125) opens a dialog with detailed information. This dialog shows the status of the selected schedule entry and the related actions.

The table rows appear in different colors depending on the execution status (for the color legend, see Chapter 7.5.1.2 page 373).

The "Edit" button is only active when the schedule entry status is "Not started" and the user who is logged in has access permissions to the corresponding project. Clicking on this button opens the dialog for editing the schedule entry (see Chapter 7.5.4 page 380).

Clicking on the "Show schedule entry log file" button opens the schedule entry's log file.

Action: the name of the action that was or will be executed.

Duration: period when the action was last executed. If the action has not yet been executed, the cell remains empty.

Errors/warnings: the number of errors and warnings that occurred during execution is displayed here.



Status: this column describes the execution status of the schedule entry (see Chapter 7.5.1.2 page 373).

Clicking on the "Show action log file" opens the log file for the selected action. The log file can alternatively also be opened by double-clicking on the desired action.



Schedule entries or actions related to an entry with the "Not started" status do not have a log file.

7.5.2 Schedule management

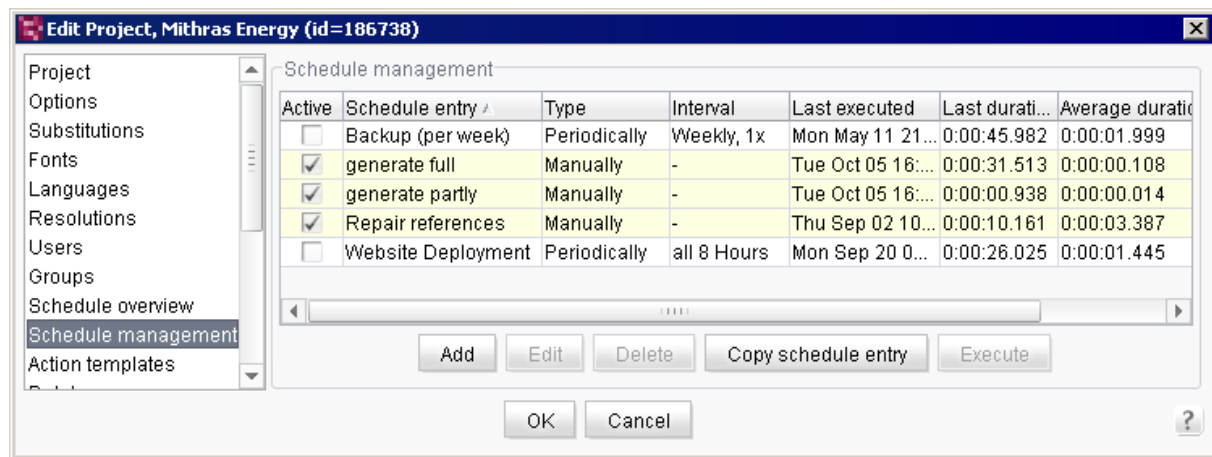


Figure 7-127: Schedule management

Schedule management contains all schedule entries that (depending on whether the user is in the server or project properties) were either created as server-based (see Chapter 7.3.9) or project-based (see Chapter 7.4.9).



Some changes to schedule entries are saved immediately without having to click on OK to confirm. This type of save is not reset if "Cancel" is used to close the dialog.

Active: if this option is *unchecked*, this schedule entry will not be subject to automatic execution. This also applies to system schedule entries (in yellow). In this case the schedule entries cannot be started in SiteArchitect, since the menu entries are grayed out; this also applies to project administrators and the server administrator. Regardless of the selection for "Interactive execution", schedule entries can be started in SiteArchitect by server and project administrators.



Schedule entry: unique name of the schedule entry. The name is defined when a new schedule entry is added.

Type: there are different types of execution methods available for schedule entries (see page 380).

- Manually: schedule entry is manually executed
- Schedule entries that are executed automatically can be distinguished as follows:
 - Once
 - Periodically

Interval: this column shows at what interval a schedule entry will be executed. This information is only displayed for schedule entries that are automatically executed periodically:

- Daily
- Every n minutes
- Weekly, n x
- Monthly, on the n day



A schedule entry can be saved only after a correct starting point can be determined. If, for instance, weekly execution is configured, a weekday must also be specified.

Last executed: this column shows the day and time the schedule entry was last executed.

Last duration: the period at which the last schedule entry was executed is displayed here.

Average duration: the average duration, as well as the "last duration" is part of the statistical information that can be useful for scheduling time.

Schedule entry ID: unique ID for the schedule entry. The ID is assigned automatically when a new schedule entry is created.

Clicking on the "Add" button creates a new schedule entry for this project or the server (see Chapter 7.5.4 page 380).

Clicking on the "Edit" button (or double-clicking on the desired schedule entry) allows the user to edit the selected schedule entry. A new dialog with the corresponding input screen opens (see Chapter 7.5.4 page 380).



Clicking on the "Delete" button deletes the selected schedule entry from the project or server. A confirmation prompt must be confirmed before deletion. The (yellow) system schedule entries cannot be deleted.

Clicking on the "Execute" button allows for immediate manual execution of the desired schedule entry, regardless of the configured execution interval.

The "Copy schedule entry" button is used to copy a schedule entry from a different project and add it to the current project or to the server-wide schedule entries. The first step is to select the project from which an existing schedule entry will be copied. The selection dialog shows all projects on the server for which the logged in user is registered as a **project administrator**.

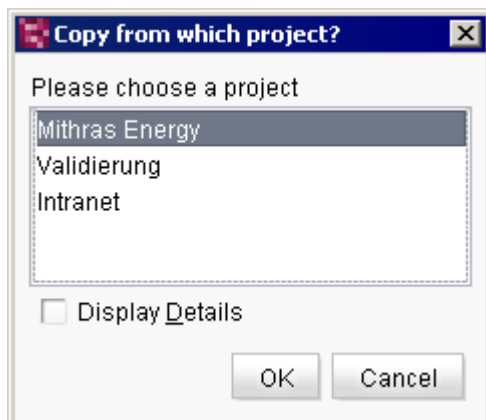


Figure 7-128: Copying a schedule entry – Selecting a project

In the next step, a new dialog appears that lists all schedule entries of the selected project.

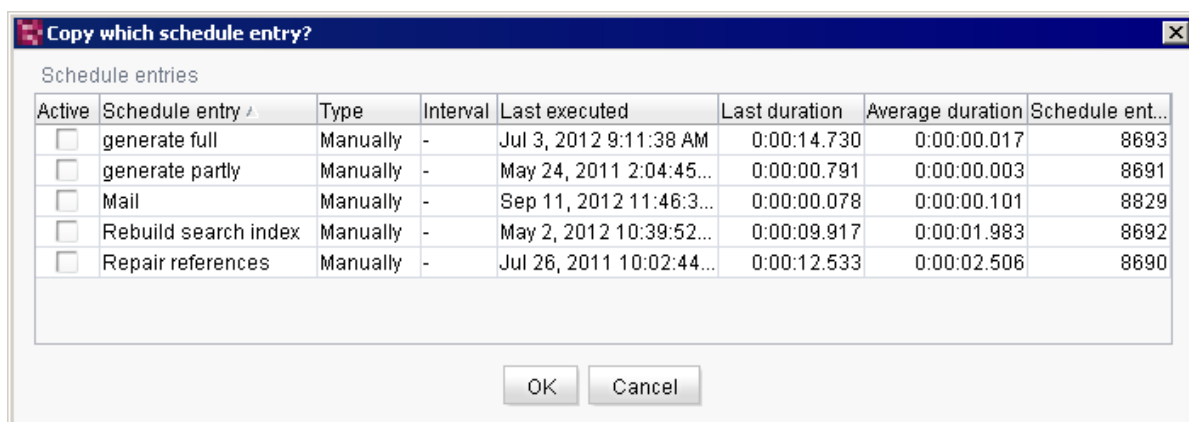


Figure 7-129: Copying a schedule entry – Selecting the schedule entry to copy

Double-clicking on a single schedule entry copies it from here to the edited schedule entry.



7.5.3 Action templates

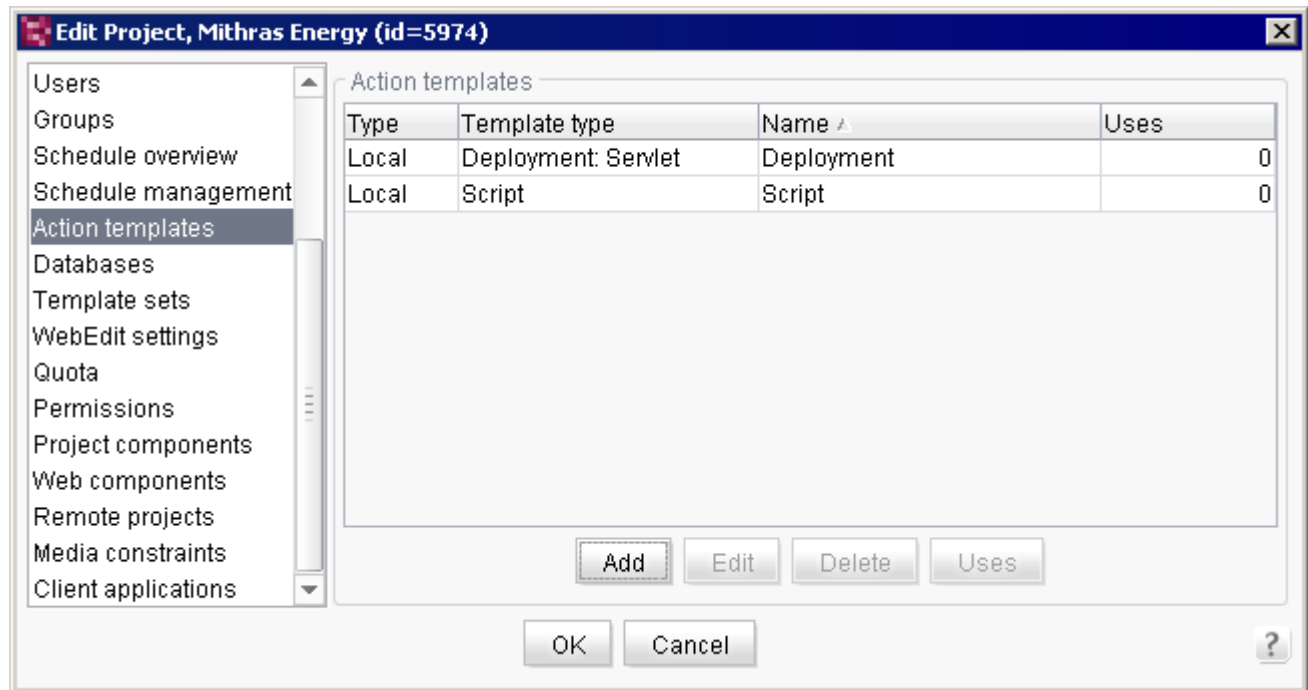


Figure 7-130: Action templates



Some changes to action templates are saved immediately without having to click on OK to confirm. This type of save is not reset if "Cancel" is used to close the dialog.

Action templates make it easy to manage actions that are to be used in multiple schedule entries – even in other projects – using identical configuration. Action templates can only be edited where they were created. To make action templates available to other projects or server-side schedule entries, these need to be marked as "public".

Public: action templates are basically always available to the project in which they were created. If this option is *selected*, the action is also available for use in all other projects.

Template type: displays the type of action. Possible options are:

- Execute deployment: see Chapter 7.5.10.6 page 413
- Execute script: see Chapter 7.5.9.4 page 395
- Send e-mail: see Chapter 7.5.9.1 page 392



Name: name of the action. The name is assigned automatically depending on the selected action (see template type):

- Deployment
- Script
- Mail

Uses: number of schedule entries that use this action template.

Clicking on the "Add" button creates a new action template. First the action type must be selected (see Chapter 7.5.6 page 386).

Clicking on the "Edit" button allows the user to edit the selected action template. A new dialog with the corresponding input screen opens (see Chapter 7.5.10 page 401 and Chapter 7.5.9 page 391).



Changes made to an action template affects all schedule entries that use this template.

Clicking on the "Delete" button allows the user to delete the selected action template. A prompt must be confirmed before deletion.



Deleting an action template is only possible if it is no longer being used in any schedule entry.

Clicking on the "Uses" button opens a new dialog that shows all schedule entries in which this action template is used:



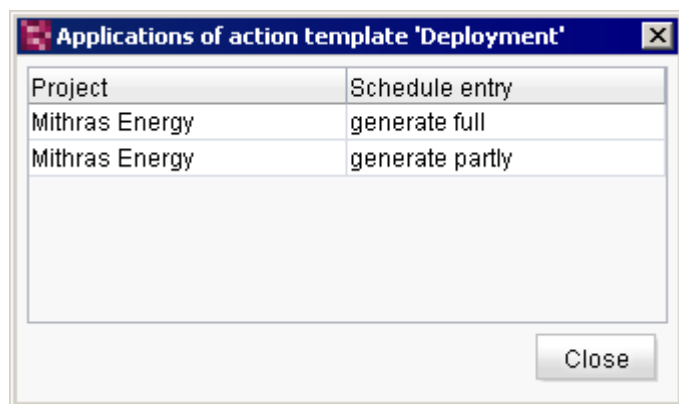


Figure 7-131: Displaying uses of a selected action template

Project: the name of the project to which the schedule entry belongs.

Schedule entry: the name of the schedule entry that uses the action template.

7.5.4 Add/Edit schedule entry (Properties tab)

Clicking on the "Add" or "Edit" button within schedule management (or double-clicking on the desired table entry) opens the "Edit schedule entry" dialog box. The following schedule entry properties can be defined on the "Properties" tab.



The screenshot shows a Windows-style dialog box titled "Schedule entry planning: Edit schedule entry". It has two tabs: "Properties" (selected) and "Actions".

Properties Tab:

- Schedule entry ID:** 8693
- schedule entry name:** generate full
- e-mail distribution list:** (empty text box)
- Directory:** mithras_full
- Execution Options:**
 - ☒ Manually
 - ☐ Once: Executed on Dec 13, 2012 at 2:21 PM
 - ☐ Periodically: First executed on Dec 13, 2012 at 2:21 PM
 - ☐ Daily: Execution rule (empty text box)
 - ☐ Weekly: Execution rule (empty text box)
 - ☐ Monthly: Execution rule (empty text box)
 - ☐ Interval: 0 Weeks 0 Days 1 Hours 0 Minutes
- Interactive execution:**
 - ☒ Interactive execution allowed for:
 - Users:** (selected tab, empty list)
 - Groups:** (empty list)
 - Parallel execution:** Not allowed (cancel)

At the bottom are "OK", "Cancel", and a help icon (?) buttons.

Figure 7-132: Edit schedule entry – Properties



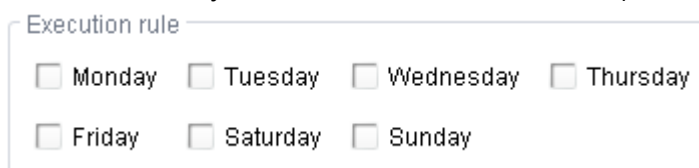
Schedule entry name: unique name of the schedule entry. The schedule entry is displayed in ServerManager (see Figure 7-125 and Figure 7-127) and in SiteArchitect using this name.

E-mail distribution list: users can specify the e-mail addresses that, among other things, can also be passed on to and used by integrated "e-mail actions" (see Chapter 7.5.9.1 page 392).

Directory: the name of the associated generation directory can be specified here. Entering this type of directory prevents overwriting in case there are multiple generation tasks.

Execution type: the radio buttons are used to select one of the following execution types. The "Manually" execution type is selected by default.

- **Manually:** if this execution type is selected, the schedule entry can only be started manually. Entering an execution date is therefore not possible. The corresponding fields are disabled.
- **Once:** unlike manual execution, one-time execution of a schedule entry is started automatically on a particular date and time. The execution time can be defined by clicking on the calendar icon.
- **Periodically:** if this execution type is selected, the schedule entry is executed automatically at regular intervals. This is done by setting the first execution time by clicking on the calendar icon and then setting the execution rule. A schedule entry can only be saved after a valid start time has been detected (this is why default values are entered, but they can be changed to any time). Periodic schedule entries can be executed according to the following rules:
 - **Daily**
A schedule entry that is executed daily does not require any additional execution rules, since it is always executed at the set time.
 - **Weekly**
In the case of a schedule entry that is executed weekly, the user must define on which weekdays it is to be executed (Monday is the default value).



Execution rule

☒ Monday ☐ Tuesday ☐ Wednesday ☐ Thursday

☐ Friday ☐ Saturday ☐ Sunday

Figure 7-133: Execution rule – Weekly execution



- Monthly

In the case of a schedule entry that is executed monthly, the user must define on which day of the month it is to be executed (the first Monday of the month is set by default).

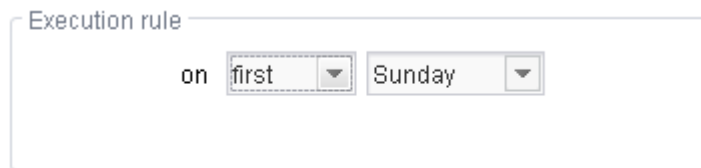


Figure 7-134: Execution rule – Monthly execution

- Interval

If the "Interval" radio button is selected, it is only necessary to specify the interval between executions (1 week is the default value).

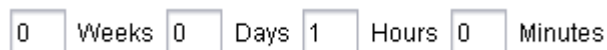


Figure 7-135: Execution rule – Interval execution

Additional schedule entry properties can be edited depending on the selected execution type (see Figure 7-132):

Interactive execution: if this option is *selected*, all selected users, or users who are members of one of the selected groups, may also execute this schedule entry interactively. It is only possible to execute a schedule entry interactively from SiteArchitect.

Users

This table lists all users who are authorized to execute this schedule interactively (from SiteArchitect).

Clicking on the folder button opens a dialog where all users of the current project are displayed. Users already selected are in bold.

Clicking on the trash can button will remove all selected users from the table. These users will no longer be authorized to execute this schedule entry from SiteArchitect.

Groups

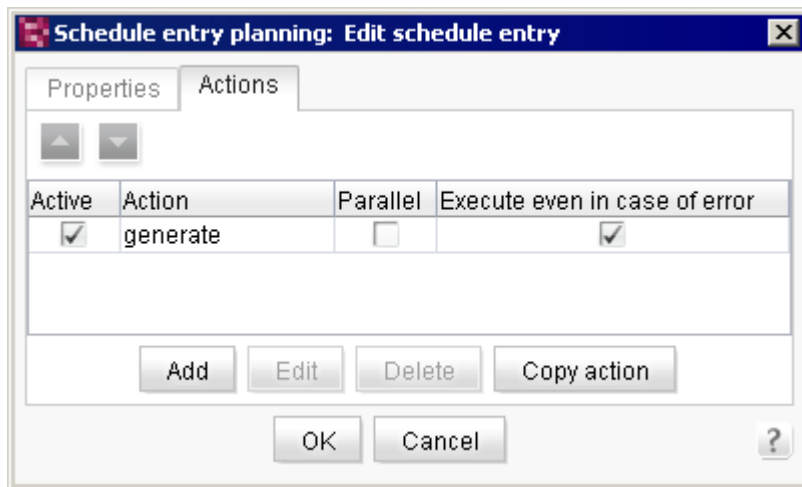
This table lists all user groups that are authorized to execute this schedule entry interactively.

Adding and deleting a group is handled the same way as adding and deleting users.





Parallel execution:

- Not allowed (cancel):
If this option is selected, parallel execution of this schedule entry is not possible. If a schedule entry is already running, any attempt to start execution will be blocked.
- Allowed (parallel execution):
If this option is selected, execution of the schedule entry will be started immediately upon request (even if a schedule entry is already running at the same time).
- Not allowed (consecutive execution):
If this option is selected, concurrent execution of the schedule entry is not possible. The schedule entry is started automatically, however, after the current execution finishes (if present).

7.5.5 Add/Edit schedule entry (Actions tab)**Figure 7-136: Edit schedule entry – Actions**

Clicking on the "Add" or "Edit" button within schedule management (or double-clicking on the desired table entry) opens the "Schedule entry planning" dialog box. Actions for the schedule entry can be added, edited or deleted on the "Actions" tab.

All related actions are displayed in the order in which they are executed during execution of the schedule entry. This order can be changed by selecting an action – in this case a row in the table – and clicking on the relevant button   to move it up or down a position.

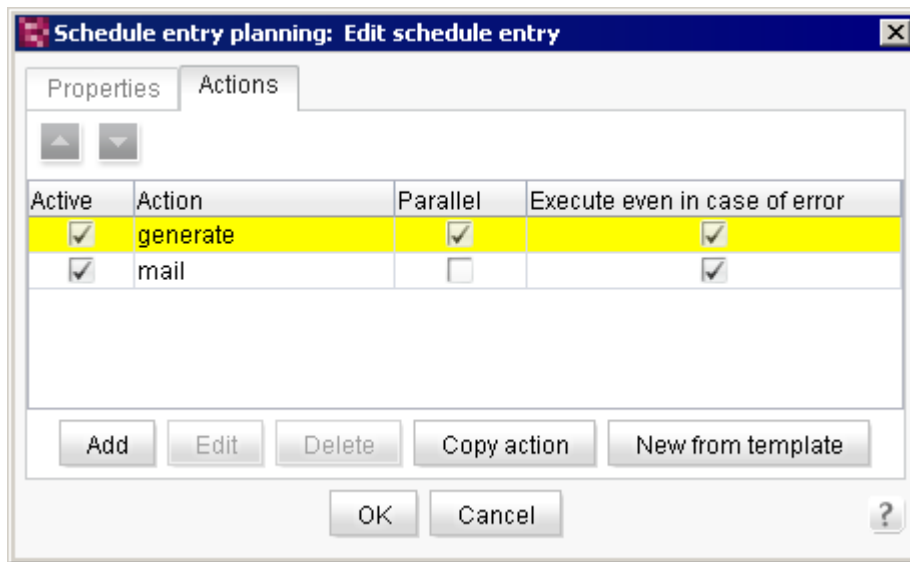
The table contains three columns: one for the name and two for action properties that are critical for execution.



Active: if the option is *selected*, this action is also executed when the schedule entry is executed; if the box is *unchecked*, the action is skipped.

Action: name of the action.

Parallel: if the option is *selected*, this action may be executed within a schedule entry concurrently with another action. This setting is only useful if multiple actions in a row are also released for concurrent execution and if these actions are logically separate from each other. If the option is selected, the rows within the overview are yellow.



Execute even in case of error: if this option is selected, after an action is executed, the next action is executed even if there were errors in the first one.

Clicking on the "Add" button creates a new action for this schedule entry (see Chapter 7.5.6 page 386).

Clicking on the "Edit" button (or double-clicking on the table entry) opens a new dialog where the selected action can be edited (see Chapter 7.5.10 page 401, and Chapter 7.5.9 page 391).

Clicking on the "Delete" button will delete the selected action from this schedule entry. A prompt must be confirmed before deletion.

The "Copy action" button is used to copy an action from a different schedule entry and add it to the current schedule entry (see Chapter 7.5.7 page 389).

Clicking on the "New from template" button allows the user to add an action that was previously defined in the action templates (see Chapter 7.5.3 page 378) to the current schedule entry (see



Chapter 7.5.8 page 390).



Actions that are added to the schedule entry using a template cannot be edited here. Changes to this action can only be made in the corresponding template in the "Action templates" area (see Chapter 7.5.3 page 378).

7.5.6 Adding actions to a schedule entry

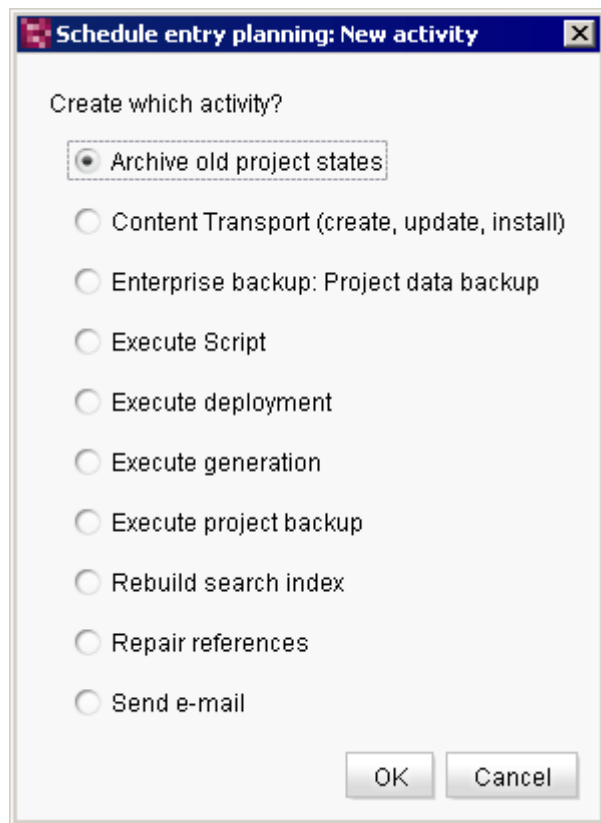
Clicking on the "Add" or "Edit" button within schedule management (or double-clicking on the desired table entry) opens the "New activity" dialog box.

As already described in the introduction in Chapter 7.5, when creating an action, a distinction is made between project-based or server-based schedule entries:

- Adding a project-based action (see 7.5.6.1).
- Adding a server -based action (see 7.5.6.2).



7.5.6.1 Adding a project-based action

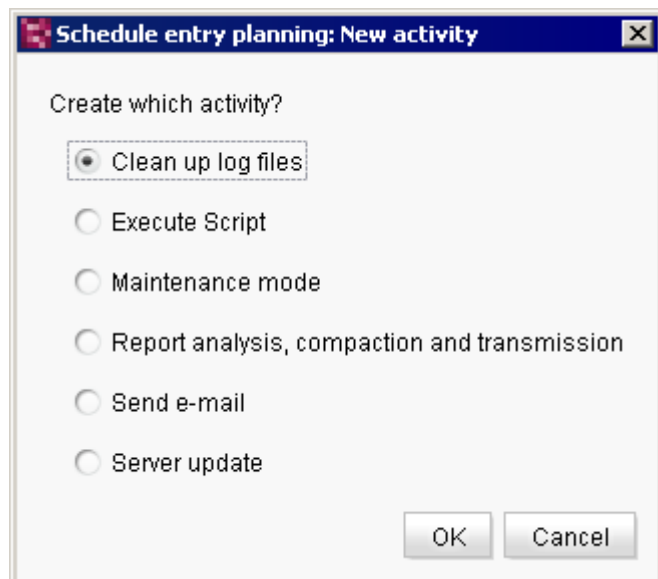
**Figure 7-137: Adding a project-based action**

The input screen of the selected action type opens based on the selected action:

- | | |
|------------------------------|-------------------------------------|
| ▪ Archive old project states | see Chapter 7.5.10.1 page 402 |
| ▪ Content Transport | see respective module documentation |
| ▪ Enterprise backup | see respective module documentation |
| ▪ Execute Script: | see Chapter 7.5.9.4 page 395 |
| ▪ Execute deployment: | see Chapter 7.5.10.6 page 413 |
| ▪ Execute generation | see Chapter 7.5.10.2 page 406 |
| ▪ Execute project backup | see Chapter 7.5.10.3 page 412 |
| ▪ Rebuild search index | see Chapter 7.5.10.5 page 412 |
| ▪ Repair references | see Chapter 7.5.10.4 page 412 |
| ▪ Send e-mail: | see Chapter 7.5.9.1 page 392 |



7.5.6.2 Adding a server-based action

**Figure 7-138: Adding a server-based action**

The input screen of the selected action type opens according to the particular selection:

- Send e-mail: see Chapter 7.5.9.1 page 392
- Clean up log files see Chapter 7.5.9.2 page 394
- Report analysis see Chapter 7.5.9.3 page 395
- Server update see Chapter 7.5.9.3 page 395
- Execute script: see Chapter 7.5.9.4 page 395
- Maintenance mode see Chapter 7.5.9.5 page 397



7.5.7 Copying actions from a different schedule entry

The "Copy action" button in the "Edit schedule entry" dialog (see Chapter 7.5.5 page 384) is used to copy an existing action from a different schedule entry to the schedule entry currently being edited. The following dialog box opens:

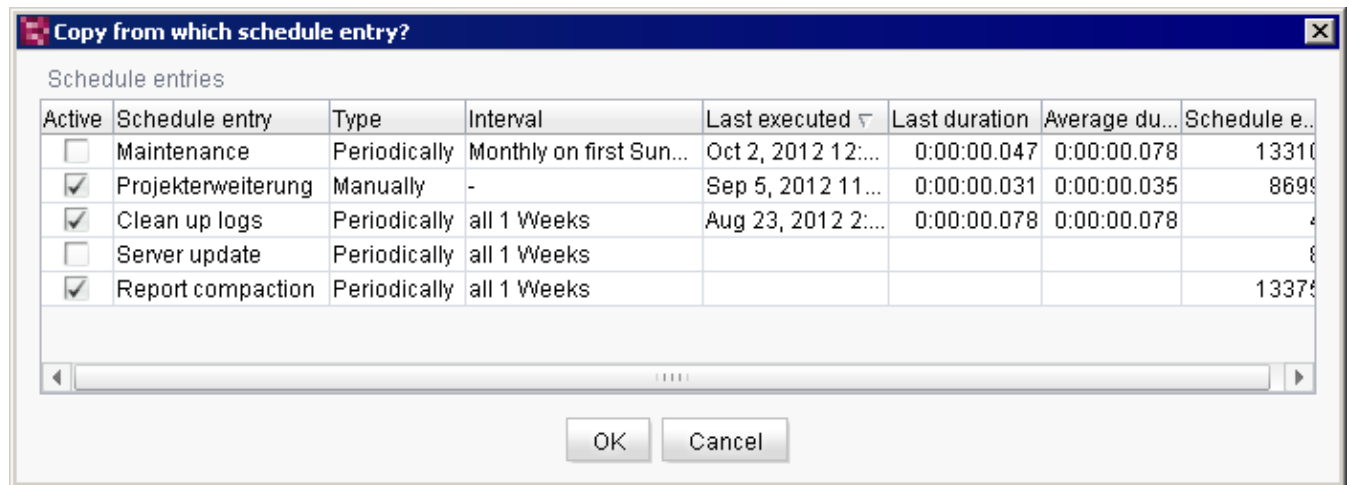


Figure 7-139: Copy action

This dialog shows a table of all existing schedule entries:

- Within project properties: all project-based schedule entries.
- Within server properties: all server-based schedule entries.

Double-clicking selects the schedule entry from which one or more actions are to be copied. A new dialog opens where all actions of the selected schedule entry are listed:

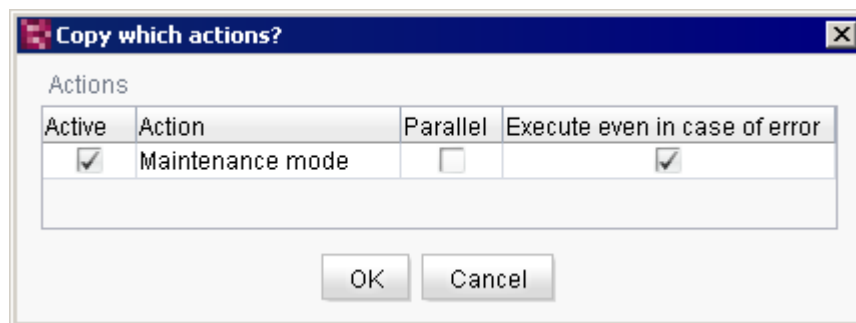


Figure 7-140: Copy action – Selecting the action to copy

Double-clicking on a single schedule entry copies it from here to the edited schedule entry.





In addition to the ability to copy individual actions, an alternative is to copy multiple actions simultaneously. To do this, select all actions by pressing the "CTRL" key. After confirming by clicking on "OK", all selected actions are copied and added to the schedule entry.

7.5.8 Inserting actions using action templates

For actions that are to be used in multiple schedule entries with the same configuration, they can be created in the same manner as in the FirstSpirit "Action templates" template concept (see 7.5.3). The action template is used to insert an action into any number of schedule entries.

An action that is based on an existing action template can be created using the "New from template" button in the "Edit schedule entry" dialog box (see Chapter 7.5.5 page 384) and then inserted into the schedule entry currently being edited. The following dialog box opens:

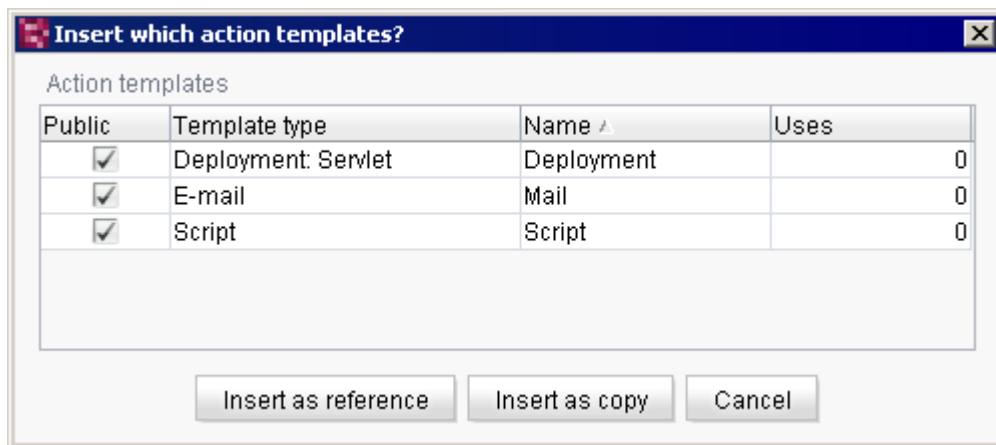


Figure 7-141: Inserting an action – Selecting an action template

Figure 7-141 shows the available action templates.

The only action templates available are ones that have been defined either in the local action template manager or, for instance, in the action template manager of a different project and that have been marked as "public" (see Chapter 7.5.3 page 378). Double-click to insert individual actions into the schedule entry.





In addition to the ability to create individual actions using action templates, an alternative is to insert multiple actions simultaneously. To do this, select all desired templates by pressing the "CTRL" key.

Clicking on the "Insert as reference" button inserts an action into the schedule entry as a reference to an action template. Actions that have been inserted as a reference to an action template can only be edited in the template manager itself and not in any schedule entry. Referenced actions are displayed in italics in the table.

Clicking on the "Insert as copy" button inserts an action into the schedule entry as a copy of the selected action template. Copies of an action can be edited as usual.

Clicking on "Cancel" closes the dialog box. No action is inserted into the schedule entry.

7.5.9 Server-based actions

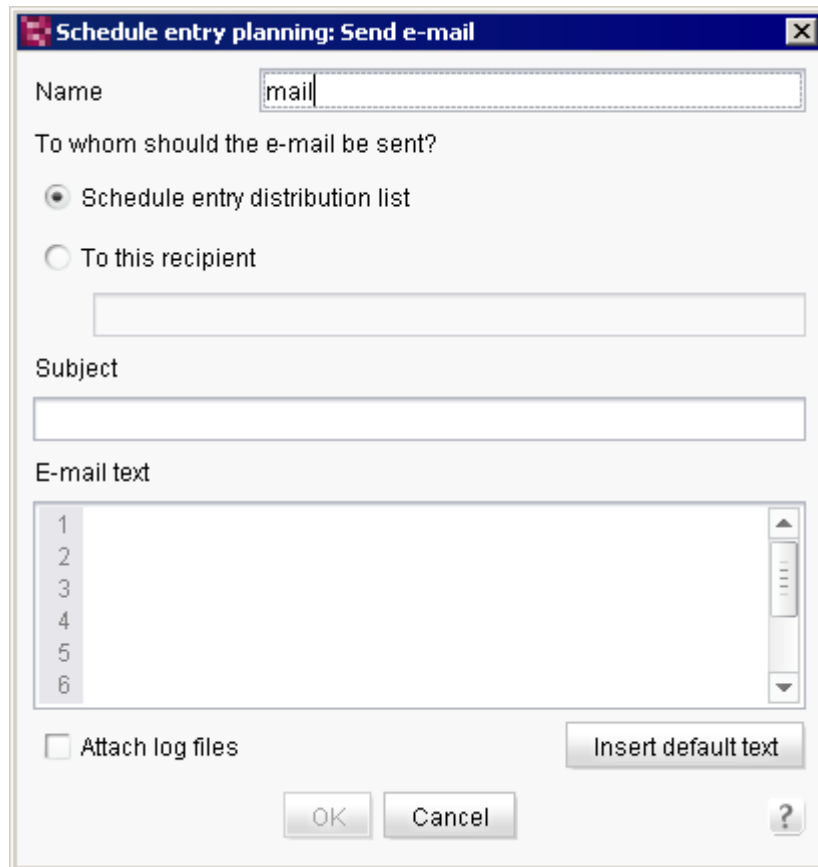
Server-based actions are created within the server properties and are inserted into server-based schedule entries.

The following actions are available:

- Send e-mail: Similar to Chapter 7.5.9.1 (see page 392)
- Execute script: Similar to Chapter 7.5.9.4 (see page 395)



7.5.9.1 Send e-mail



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:

1
2
3
4
5
6

☐ Attach log files

Insert default text

OK Cancel ?

Figure 7-142: Create action – Send e-mail

This action offers the ability to send e-mail messages. The user can use either the e-mail distribution list of the associated schedule entry or create a custom distribution list. The unique feature here is that it is possible to include in the attachment previously executed actions in the same schedule entry.

Schedule entry distribution list: if this option is *selected*, the e-mail is sent to all recipients defined in the distribution list of the associated schedule entry.

To this recipient: if this option is *selected*, the e-mail will be sent to all e-mail addresses defined in the following text box.

Subject: the subject of the e-mail to be sent.

E-mail text: the body of the e-mail is entered here. This text may contain template syntax, which will then be parsed before the e-mail is sent. The `#context` variable is used to output



information on the current schedule entry (including the actions executed in this schedule entry) and the `#task` variable is used to output information on the "Send e-mail" action³³.

Examples:

`$CMS_VALUE(#context.getStartTime())` \$: outputs the start time of the schedule entry

`$CMS_VALUE(#task.getSubject())` \$: outputs the subject of the sent e-mail

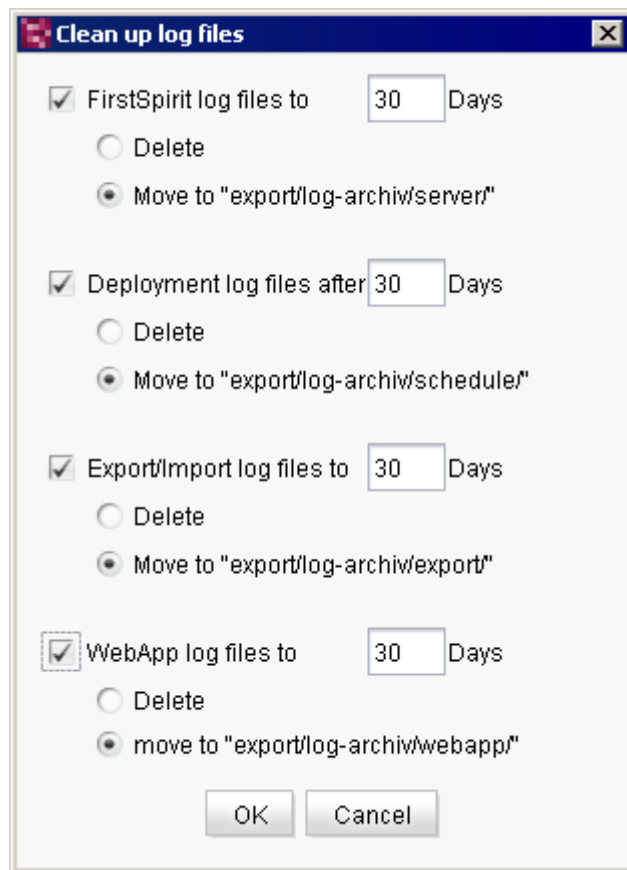
Attach log files: if this option is *selected*, the log files (if they exist) of the previously executed actions from the same schedule entry are sent in the e-mail as an attachment.

The "Insert default text" button is used to insert the content of the configuration file for the standard template (`%serverdirectory%/conf/server/DefaultMailText.txt`) into the current e-mail body text.

³³ Methods for `#context` and `#task` can be found in the FirstSpirit Access API (*de.espirit.firstspirit.access.schedule.ScheduleContext* or *de.espirit.firstspirit.access.schedule.MailTask*).



7.5.9.2 Clean up log files

**Figure 7-143: Create action – Clean up log files**

To keep the server log files clearly organized, the user has the option of cleaning them up using this function. The user can specify the age of the log files to be taken into account when performing this action. Depending on the selection, the files will be deleted or moved to a relevant archive directory. Selecting the log files to clean up can be controlled by checking or unchecking the respective checkboxes.

An activated schedule entry with this action, which automatically runs once a week, is already added by default after the server is installed.



7.5.9.3 Report analysis, compaction and transmission / server update

These two features are not yet authorized for use by end customers in FirstSpirit Version 5.0 and are still available only to partners for beta testing.

Updating the server manually, however, is possible through ServerMonitoring.

See Chapter 8.6.2.3 page 480 for more information on this.

7.5.9.4 Execute script

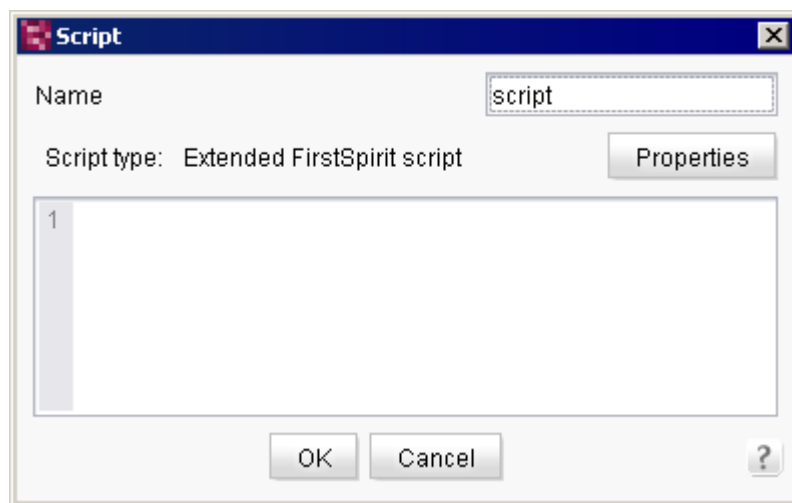


Figure 7-144: Create action – Execute script

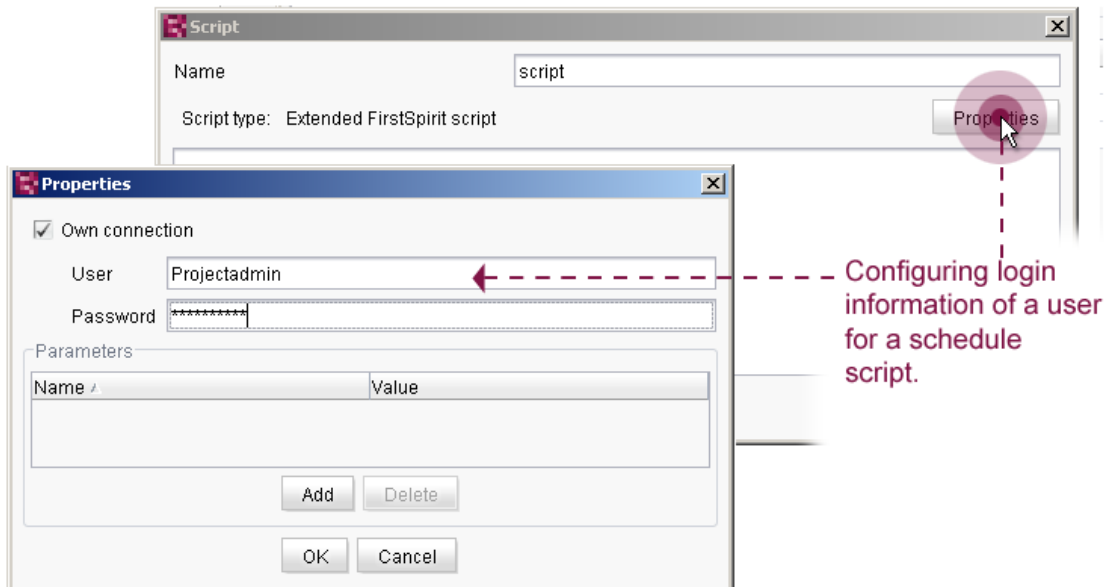
Name: a new name for the script can be specified in this field.

Text input field: the script code to be executed in this action is entered in this field.

Clicking on the "Properties" button opens a new dialog where the properties of this script can be edited (see Chapter 7.5.9.4.1 page 396). This is important, for instance, if the script is to carry out change operations in the project (e.g. create or modify objects).



7.5.9.4.1 Script properties

**Figure 7-145: Editing a script – Script properties**

If required, this dialog can be used to establish its own connection to the server instead of using the current session to execute scripts.

Own connection: if this option is *selected*, a new server connection is established using the following data for executing the script.

User: the user name to be used to log on to the server.

Password: the password corresponding to the user specified above.



Establishing a new server connection requires the use of an internal FirstSpirit user's data. It is not possible to use an external user.

Parameters: all parameters are entered here that are to be taken into account when executing the script.

If the **Own connection** box is checked, the editorial permissions of the user specified here are evaluated and taken into account when executing the script.

It is possible to execute schedule entry scripts via the system connection, e.g.:

```
context.getUserService();
```



The variable `context` provides a special connection that only offers read-only access (via the `CAN_SEE`, `CAN_READ`, `CAN_META_SEE` permissions). This type of connection is used by scripts that have been configured without special user information. Only the operations that do not make changes can be executed using them. Otherwise, a security exception will cancel execution of the script.

If change operations are to be carried out in the project using a schedule entry script, the schedule entry script must use the specified user's login information. This user must have the relevant permissions (e.g. `CAN_CHANGE`, `APPEND`, `DELETE`, etc.) and must first be defined via the script properties (see above). A user-specific connection (based on the stored user information) must then be made within the script using the `connection` variable, e.g.:

```
connection.getProjectById(context.getProject().getId()).getUserService();
```

The schedule entry script then runs based on the specified user's permissions and evaluates its editorial permissions (e.g. for creation) for every changing operation.



If change actions in a script (within the schedule entry) are executed on project properties, this script has to be executed based on either the user's server or project administrator permissions.

7.5.9.5 Maintenance mode

Maintenance mode is used for the following purposes:

- To update the FirstSpirit server. The server must be shut down to do this. (However, this is not carried out automatically by maintenance mode functions; it must be carried out manually while in maintenance mode.)
- To limit access to projects (e.g. for conversions, larger updates), even in the case of particular user groups, if necessary.

If maintenance mode is activated, depending on the configuration, no users can log on to the FirstSpirit server or to selected projects; open clients are automatically terminated, depending on the configuration. Users logged onto FirstSpirit are notified that the server will be shut down and will thus be able to save their current changes before logging out.



To activate maintenance mode, a corresponding server schedule entry is required.

Maintenance mode

Notification to display

☒ Display notification on web page for all projects

Period durations (each period begins as the previous expires):

1. Display notification after minutes

2. Show End Sessions warning after minutes

3. Refuse new sessions after minutes



4. Start maintenance mode after minutes



5. Estimated duration minutes

☐ End sessions during maintenance, refuse new sessions

☐ Refuse to start tasks

☒ Apply to all projects

Apply to projects  

Do not apply to users  



Do not apply to groups  

Figure 7-146: Create action – Maintenance mode

All information within the dialog except for the maintenance mode start time (item 4) and the planned duration (item 5) are optional. If this information is present, make sure that it is compatible with the rest of the information included here.

Notification to display: text can be entered in this field that will be output in addition to the system messages.

Display notification on web page for all projects: this option is selected by default. If it is selected, the system messages are displayed only in the clients and not on the FirstSpirit start page.



Maintenance mode comprises multiple levels. Items 1 through 5 are processed automatically in consecutive order as soon as the schedule entry is started. The information in minutes specifies the start times of the individual stages in relation to the previous stage. If 0 is entered for the minutes or the field is left empty, the particular stage is skipped. Figure 7-146 shows the default settings. The output system messages depend on whether the schedule entry applies to the entire server (option: "Apply to all projects", see below) or only to one or more projects (option "Apply to projects", see below).

1. Display notification after: based on the number entered here in minutes, the system message "Maintenance work will be carried out on FirstSpirit Server | on this project in x minutes. Estimated duration: approx. y minutes." plus any text input in the "Notifications to display" field is output as an advanced warning that maintenance work is scheduled. x is the number entered in this field; y is the number entered for the "Estimated duration" field.

2. Show End Sessions warning after: based on the number of minutes entered here, the system message "FirstSpirit Server | Project is unavailable due to maintenance work starting 21.08.2012 15:15:00. Estimated duration: approx. y minutes. Please terminate your session." plus any text input in the "Notifications to display" field is output. Users who are logged in now have the opportunity to save their work and log out. The date and time are the start time for the maintenance mode schedule entry plus the stage 1 through 4 information.

3. Refuse new sessions after: if the option "End session during maintenance, refuse new sessions" (see below) is selected, no new logins will be possible on the server or in the selected projects after the time specified here in minutes.

4. Start maintenance mode after: the maintenance period begins after the number of minutes specified here. If the option "End sessions during maintenance, refuse **new sessions**" (see unterhalb) is selected, clients still logged in will be terminated upon receiving the message "FirstSpirit Server | Project is not available due to maintenance work. Your session has been terminated.". Depending on the setting, the system message on the start page and in the client appears as follows: "FirstSpirit Server has been in maintenance mode since 21.08.2012 15:15:00. Estimated duration: approx. y minutes.". Maintenance work can now be performed and, if necessary, the FirstSpirit server can be shut down.

5. Estimated duration: the number of minutes estimated for maintenance can be entered here. After this time span has elapsed, the following system message will appear automatically: "Maintenance work on FirstSpirit Server | on project has been completed." Users can then log back onto the server. If while a schedule is running it is discovered that maintenance mode is needs to run longer than is set here, the schedule (before this stage finishes) should be terminated manually.



ServerMonitoring can be used to view the stage in which a maintenance mode schedule is running (see Chapter 8.6.2.1 page 477).

End sessions during maintenance, refuse new sessions: if this option is selected, all server or selected project sessions are terminated during the maintenance period. If this option is not selected, no sessions will be terminated. To allow new sessions, at stage 3, "Refuse new sessions after", the value 0 must be set.

Refuse start tasks: if this option is selected, schedule entries that take place during the maintenance period are canceled.

Apply to all projects: if this option is activated, the schedule applies to all projects on the server. If the option is not selected, the desired project(s) can be selected under "Apply to projects".

Apply to projects: here is where project(s) can be selected for which the schedule is to apply.

Do not apply to users: here you can select a user who can log in to the FirstSpirit server or the selected projects while in maintenance mode. The server administrator can always log in.

Do not apply to groups: here you can select groups that can log in to the FirstSpirit server or the selected projects while in maintenance mode.

The schedule is started for the set period or it can be started directly using the "Execute" button. Maintenance mode schedules that are already running can be terminated using FirstSpirit ServerMonitoring (see Chapter 8.6.2.1 page 477). Maintenance mode schedules that have not started yet can be terminated automatically by changing the setting to "Manually".



7.5.10 Project-based actions

Project-based actions are created within the individual project's project properties and are added to project-based schedule entries.

The following actions are available:

- Archive old project states see Chapter 7.5.10.1 page 402
- Content Transport see respective module documentation
- Enterprise backup see respective module documentation
- Execute Script: see Chapter 7.5.9.4 page 395
- Execute deployment: see Chapter 7.5.10.6 page 413
- Execute generation see Chapter 7.5.10.2 page 406
- Execute project backup see Chapter 7.5.10.3 page 412
- Rebuild search index see Chapter 7.5.10.5 page 412
- Repair references see Chapter 7.5.10.4 page 412
- Send e-mail: see Chapter 7.5.9.1 page 392



7.5.10.1 Archive old project states



To compare the use of the archive function to using the "FirstSpirit EnterpriseBackup" module, also refer to FirstSpirit Release Notes 4.2, "Long-term archiving and backup in FirstSpirit".

Project archiving

Version history

- ☐ Maintain version history completely (no archiving)
- ☒ Maintain version history at least day(s) (partial archiving)
- ☐ Do not maintain version history (complete archiving)

Objects

- ☒ Content, media and data content
- ☒ Templates
- ☐ System data

Options

- ☐ archive only objects marked as deleted
- ☒ archive deleted objects and version history that is no longer required
- ☐ Waiting time per archiving step ms

Runtime

- ☐ Limit maximum archiving runtime per run to minutes
- ☒ Do not limit archiving runtime

OK Cancel

Figure 7-147: Create schedule entry – Project archiving

FirstSpirit uses repositories to archive and maintain version histories of project data. Each project has a repository in the server directory `data\projects\`. For each action made in SiteArchitect, data is written to the repository. This applies to actions that create new elements as well as actions that delete elements. In addition, deleted elements are not removed from the repository. Since new data is always being added, the repository will continue to grow and will always require more hard disk space.



The "Archive old project states" is used to carry out archival of the selected project so that data which are no longer required are moved out of the project and into a repository, reducing load times and increasing the performance of the FirstSpirit server. Data are moved from the repositories to archive files. Archive data that are no longer required can subsequently be deleted in order to free up hard disk space.

Each project has its own archive folder in the server directory `archive` and the data to be archived are moved to an archive file (`tar.gz` format) in the corresponding project folder. However, they can be viewed later using the "Archive" function in the server and project properties and can be installed as needed (see Chapter 7.2.3.8 page 229).

For more information on the FirstSpirit archival concept, see Chapter 7.9, starting on page 443.



Use the archive function with caution. Depending on the configuration, the version history may no longer be complete after archival. Thus, even older revisions may no longer be restored properly. During an export (see Chapter 7.2.3.3 page 224), only the currently available project status is exported without any existing archives or archived project statuses after archiving.

The archiving criteria can be specified using the following options:

Version history:

The user can specify in this area the period of time when archival should take place or the period of time the version history should be preserved.

Maintain version history completely (no archiving): if this option is selected, archiving is not carried out and no archive file is created. The version history is preserved in its entirety.

Maintain version history at least 120 day(s) (partial archiving): this option is used to specify the period of time for which the entire version history with all revisions should be preserved. If, for instance, 120 days is set, archiving only takes into account revisions that are **older than** 120 days at the start of the archive schedule entry. All changes that are **more recent than** 120 days at the start of the archive schedule entry can also be completed without a gap even after archival.



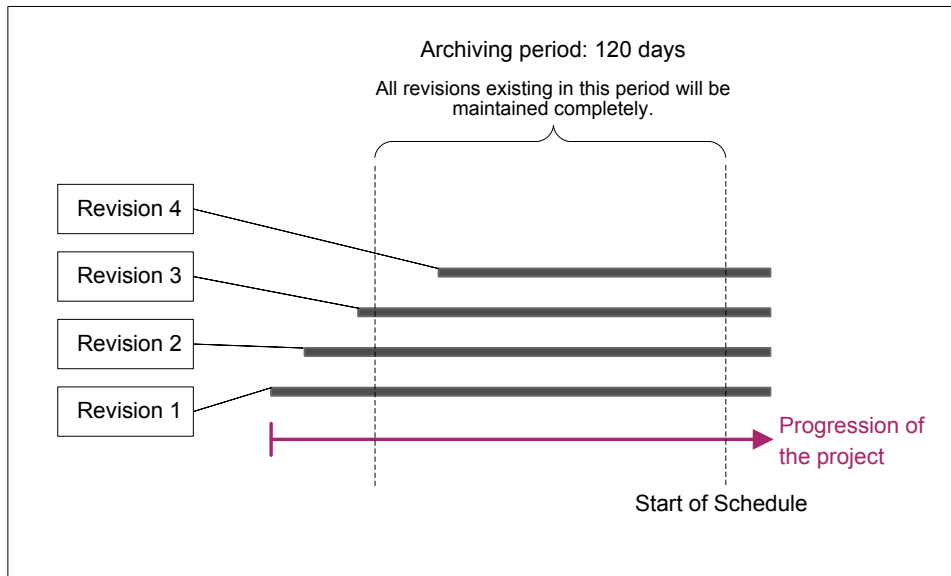


Figure 7-148: Partial archiving over 120 days

Do not maintain version history (complete archiving): if this option is selected, the entire version history is taken into account when the schedule entry is executed. All data that are no longer required (see Chapter 7.9.3 page 444 for more information) are moved to the archive file.

Objects:

The user can specify in this area what type of data is to be archived.

Content, media and data content: if this option is selected, all content from a project's page, media and data stores is archived (that is, everything except templates).



Please note that there are size limitations when archiving database content: If the amount of all database entries is larger than 90% of 8 GB the archive process for the database entries will be cancelled.

In this case a WARN will be logged (in the file fs-server.log, e.g.

```
WARN 09.08.2010 12:50:42.080 {seID=369117}
(de.espirit.or.impl.AbstractSessionHandler): cleanup for entityType='cases' aborted, tar
entry size limit reached! [schema=P222005_222001]
```

Execute further archiving schedules for archiving the remaining database entries.

Templates: if this option is selected, templates are archived. This option can only be selected in conjunction with "Content, media and data content".



System data: system data are information generated by the system for each action in SiteArchitect (e.g. creating or deleting objects, releases, etc.) (see also Chapter 7.9.1 page 443, and Chapter 7.9.2 page 443). If this option is selected, the data of closed workflows and internal revision media are archived in addition to system data that are not longer used. Revisions for which user data no longer exist are deleted completely (i.e. all internal files belonging to this revision). This also includes the UI files. Archived revisions can later be viewed using the "Archive" function on the "Revisions" tab (see Figure 7-18).

If closed schedules are also to be archived, the boxes "Content, media and data content" **and** "System data" must be checked. In this case, all files are archived that are associated with a schedule entry that was closed at a particular point in time within the archival period.

Options:

The user can specify in this area whether only objects or also version history entries that are no longer needed are to be archived.

archive objects marked as deleted only: if this option is selected, only objects that were deleted are archived.

archive deleted objects and version history no longer required: if this option is selected, in addition to deleted objects, version history entries that are no longer required are archived as well. This means that the version history of objects is reduced. In any case, however, the full version history from the period of partial archiving (as long as this option is selected), the revisions from the last release state (if present) and the current edited state are preserved.

Waiting time per archiving step: when archiving a high volume of data, this value can be used to set a pause (in milliseconds), which is added between each step in the archival process. This reduces the load on the server during the archival process.

Runtime: since archiving a high volume of data can take a long time, a maximum runtime limit for archiving a project can be set. Users can specify the amount of time allowed for archival in the "Runtime" area.

Limit maximum archiving runtime per run to 60 minutes: the user can set the maximum number of minutes for archival before the process is stopped. The default setting is 60 minutes. The next time the action is started (manually or automatically), archiving begins where it left off when last stopped. An archive file is created specifically for each of these "partial archives".

Do not limit archiving runtime: if this option is selected, archiving will continue without any time limit until it is completed.



7.5.10.2 Execute generation

Schedule entry planning: Execute generation

Properties Extended

Name: generate

☒ Perform FullGeneration

☐ Execute PartialGeneration for following start nodes

Start points

ID

Add Delete

☐ 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: Advanced URLs

Prefix for absolute paths:

Successful only if: No fatal errors and remained within threshold values.

Threshold value for normal Errors: 0

Threshold value for Warnings: 0

OK Cancel ?

Figure 7-149: Create action – Execute generation

This action is used to carry out full or partial generation of the selected project (see also Chapter 8.4 page 466).

Full generation generates all content of the project, and partial generation generates only the selected "start nodes" and their child nodes.



For further information about the generation in FirstSpirit please see also *FirstSpirit Online Documentation*, "Advanced topics" / "Generation".

"Properties" tab

Perform FullGeneration: if this option is *selected*, the project is generated in its entirety when the schedule entry is executed.



In addition to the time-consuming full generation of a project you can also determine the changed content of a project (in relation to the last successful generation) using the FirstSpirit Access API and generate only this changed content ("Delta generation"). For a comparison of the generation types and a configuration example for delta generation see FirstSpirit Online Documentation (Advanced topics/Generation)).

Execute PartialGeneration for following start nodes: if this option is *selected*, only the nodes shown in the following "Start points" (start nodes) table are generated (including all subordinate start nodes):

☒ Execute PartialGeneration for following start nodes

Start points

	ID
Homepage	7063
Startpage	6106
Press (homepage)	7713

Figure 7-150: Create action – Executing partial generation

Start points: this table shows all start nodes to be generated when the action is executed. The selected elements are performed recursively; for instance, if the Media folder is added, all subordinate elements (Media and other folders) are also considered part of the generation schedule.

Clicking on the "Add" button opens a dialog displaying all of the project's available and not yet selected start nodes. Only released objects are displayed.

Clicking on the "Delete" button will delete the selected start node.



Can be defined by user? (for interactive schedule entries only): the option "Can be defined by user" can be selected if it is a partial generation. The project administrator can allow the user to specify custom start nodes for generation when the schedule entry is executed by selecting this option. These changes only affect the current generation schedule.

Start nodes for media generation can also be defined within a generation schedule entry.

Background: when deploying project content, referenced media, such as that within an image input component on a page, are also generated and deployed. Media that are not explicitly referenced are not automatically included during generation. In some applications, for instance, if media are used within a script, these media must also be included in the generation schedule.



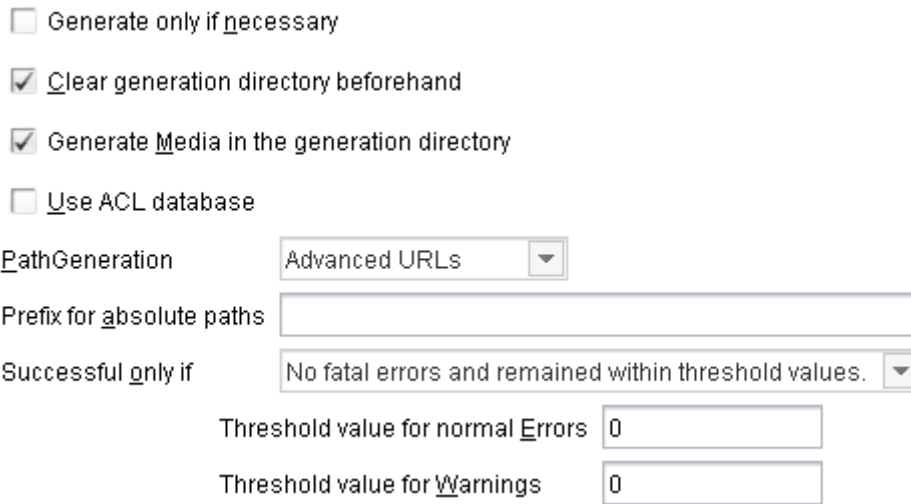
Media for which the option "Parse file" is selected (see the FirstSpirit Manual for Editors (SiteArchitect)) are not generated/deployed if the option "Copy all media to this folder during generation" is selected for the media.

The generation of media can be assigned to individual generation schedules. The purpose of the new option is to make it possible to deploy individual sub areas faster. Saving the processor intensive media generation into separate generation schedule entries improves performance, especially when executing multiple small partial generation schedules, since for each generation only the desired media are included. However, this option is also ideal for deploying media in a remote project ("remote media"), for instance.



Media for which the option "Parse file" is selected (see the FirstSpirit Manual for Editors (SiteArchitect)) are not generated/deployed if they are under the selected start node and are not referenced on any page of the generation.





☐ Generate only if necessary
☒ Clear generation directory beforehand
☒ Generate Media in the generation directory
☐ Use ACL database
 PathGeneration Advanced URLs ▼
 Prefix for absolute paths
 Successful only if No fatal errors and remained within threshold values. ▼
 Threshold value for normal Errors
 Threshold value for Warnings

Figure 7-151: Window detail (see Figure 7-149)

Generate only necessary: if this option is *selected*, the system checks if any changes were made to the project since generation was last executed before this generation is started. Generation only takes place if a new revision was generated in the project (cf. also delta generation).

Please note that it is not practical to use this option together with manual generation of a revision before generation is started (required in some configurations).

Clear generation directory beforehand: if this option is *selected*, the generation directory is cleared before generation begins.

Generate Media in the generation directory:

if this option is *selected*, all referenced media are generated automatically in the generation directory during the generation process.

If the option is *not selected*, no media are generated in the generation directory during the generation process (exception: parsed media). This may reduce the I/O load and the amount of disk space required in the generation directory. This option does *not* prevent media from being transmitted, since when internal FirstSpirit deployment mechanisms (in the file system, by FTP or by CRC, see Chapter 7.5.10.6 page 413) are used for deployment, all referenced media that are not in the generation directory are streamed over the Berkeley DB back end. This does not apply to other, external transmission mechanisms (e.g. via rsync, see Chapter 10 page 522). (Note: if this option is not selected, it is not possible to browse through the generation directory (fs5staging).)

Use ACL database: if this option is *selected*, information is stored during generation in a local database called the FirstSpirit Access Control Database (ACL database) for each page reference in the site store and for each medium in the media store. This database is used to provide



information on FirstSpirit objects, such as access permissions that were stored for an object. The Access Control Database is automatically synchronized with the currently released project status when the content is generated.

For more information on the ACL database, see the "Documentation for the FirstSpirit security module".

PathGeneration: a method for path generation can be selected from this list. There are currently four different generation types:

- Advanced URLs: for this method, the display names of the FirstSpirit objects are used as the basis for URL generation (see also *FirstSpirit Online Documentation* (Advanced topics/Generation)).
- Default URLs: for this method, each language of the project has its own subdirectory on the web server (de, en, etc., see also *FirstSpirit Online Documentation* (Advanced topics/Generation)).
- Multiview URLs: for this method, there is no language-specific subdirectory; instead, the files for each language are labeled with the respective language code. In this case, the language code is appended *after* the file extension (e.g. index.html.de, index.html.en) (see Chapter 7.4.2).
- Infix URLs: there is also no language-specific subdirectory for this method; instead, the files for each language are labeled with the respective language code. In this case, the language code is appended *before* the file extension (e.g. index.de.html, index.en.html).

Prefix for absolute paths: the prefix entered here is set as the default for all links to which the property of an absolute link is assigned in a template (in SiteArchitect).

Successful only if: there are two options: one decides when generation is successful and the other when it is not:

- No fatal errors and remained within threshold values
- No fatal errors

In both cases, generation is canceled as soon as a fatal error occurs. If no fatal errors occur, generation using the option "No fatal errors" always applies, and the "No fatal errors and remained within threshold values" option is only considered successful if the following threshold values were not exceeded:

- Threshold value for normal errors
- Threshold value for warnings



"Extended" tab**Figure 7-152: Execute generation – "Extended" tab**

Template sets: In this area you can define for which template set in which language a generation is to be executed. See also Chapter 7.4.5 page 307 (languages) and Chapter 7.4.13 page 326 (template sets).

Variables: Use this field to set variables in the generation schedule. These variables will be taken into account while generating the project. They are available in the whole project so that they can be accessed in page and section templates for example. In addition, variables which are defined in generation schedules can overwrite for example variables that are defined in the project on menu levels. See also *FirstSpirit Online Documentation*, "Template development" / "Variables" / "Contexts" / "project-related".

Variables with name and value can be defined for the generation schedule by means of the button "Add". Already defined variables can be removed by using the button "Delete". For more information about the usage of variables in FirstSpirit (definition, output, inheritance, overwriting)



see also *FirstSpirit Online Documentation*, "Template development" / "Variables".

Clustering: see Chapter 7.6.4 page 425.

The `#global.scheduleContext` call can be used to access the execution context of a generation schedule entry and thus to access information such as the project name, schedule entry start time, generation directory path, number of errors during generation, etc.

The `de.espirit.firstspirit.access.schedule.ScheduleContext` interface, which provides the necessary methods, is available in the FirstSpirit Access API.

7.5.10.3 Execute project backup

The project backup function is used to export the current state of the project. The project backup is created automatically in the path defined as the backup path in the `fs-server.conf` configuration file. (In this case it is possible to transfer the directory to a different hard disk for backup (for `BACKUP_PATH` parameter information, see Chapter 4.3.1.15 page 62).) The function "Clean up server" is used to delete export files that are no longer required for the project backup (see Chapter 7.2.2.1 page 215).

The license-dependent "FirstSpirit Enterprise Backup" module allows for efficient incremental and differential data backups. All changes to a project are backed up individually starting from a defined starting point (snapshot). If necessary, a full backup can be created from the initial backup file (snapshot export) and the respective changed backup files.

For more information, see "Documentation for the FirstSpirit backup module".

7.5.10.4 Repair references

If there are incorrect references in a project, a recalculation of the references in the project can be started using this action. After starting the project schedule entry, the message "Schedule entry completed successfully." appears. The reference calculation at this point in time was already started, but does not have to be completed yet. Calculating the references takes place in the background and can take some time depending on the size of the project.

7.5.10.5 Rebuild search index

The elements from the FirstSpirit stores (e.g. pages, sections, media files) are written to or removed from the search index during creation, editing or deletion. This action is used to rebuild the search index. It may take some time to rebuild, depending on the project size. After



performing this action, the message "Schedule entry completed successfully." appears.

The re-indexing of multiple projects is processed sequentially.

For more information, see Chapter 9.18 page 516.

Note: Besides the option to create the complete search index for a project anew, the search index can be updated also only for single sub-segments of a project. The relevant methods are made available in the FirstSpirit Access API and can be executed by means of a script (starting point FirstSpirit Access API: interface ProjectStorage (package: firstspirit.access.admin)).

7.5.10.6 Execute deployment

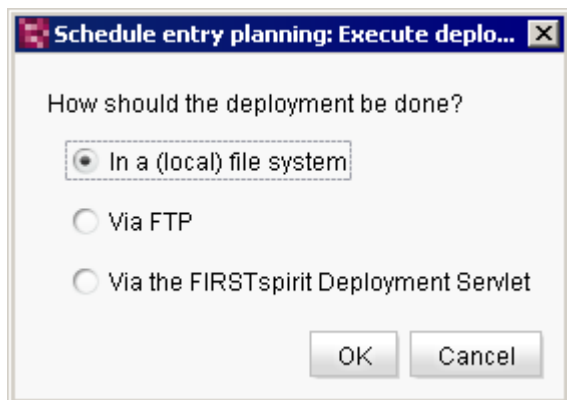


Figure 7-153: Create action – Execute deployment

The deployment type needs to be selected first so that a new deployment can be executed. There are currently three different options:

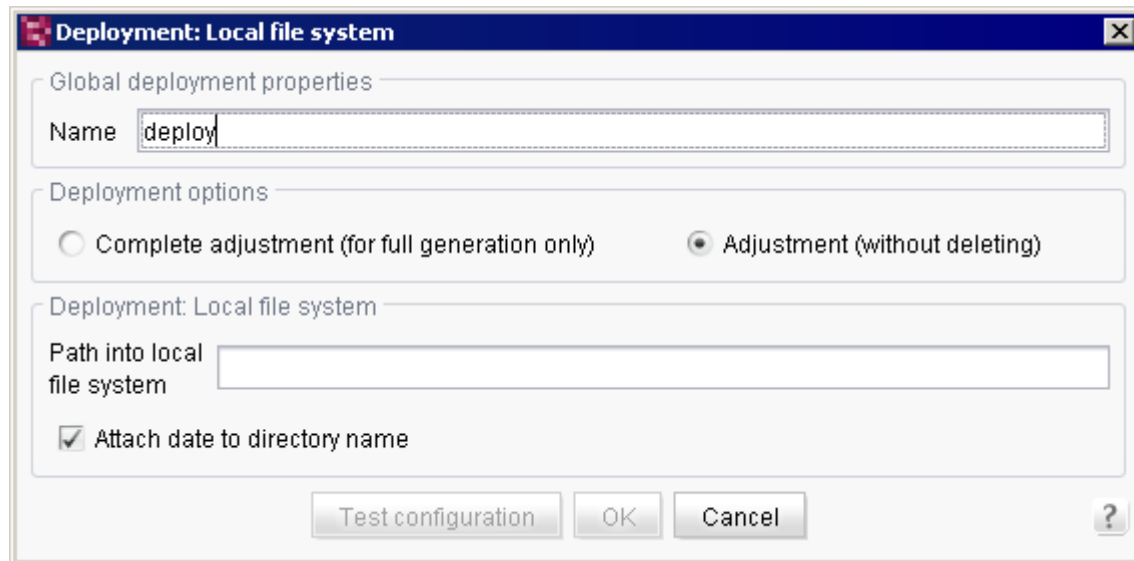
- Deployment in a local file system (see Chapter 7.5.10.6.1 page 414)
- Deployment via FTP (see Chapter 7.5.10.6.2 page 416)
- Deployment via the FirstSpirit Deployment Servlet (see Chapter 7.5.10.6.3 page 419)



If the user is not creating a deployment, but is instead editing an existing one, this dialog is skipped and the user is taken directly to the corresponding input screen based on the deployment type selected.



7.5.10.6.1 Deployment in a local file system

**Figure 7-154: Create action – Deployment in a local file system**

This action is used to deploy a project in a local file system and therefore does not require many settings.

Name: the name of the action displayed in the schedule overview, schedule management and action templates.

Deployment options**Complete adjustment (for full generation only)**

This method is used to create a data inventory identical to the generation data inventory on the web server. This means that files that are no longer present on the development server are also deleted on the web server, new files are copied, and old files already present on the server are overwritten by the new ones.

Adjustment (without deleting)

This deployment option is similar to the complete adjustment, but differs in the fact that no files are deleted from the web server, even if they are no longer present on the development server.

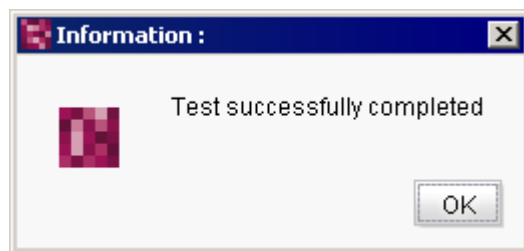
Deployment

Path into local file system: here the user can specify in which local directory deployment is to take place. The path can be specified as absolute or relative (to the server's working directory). The path information also activates the OK and "Test configuration" buttons.

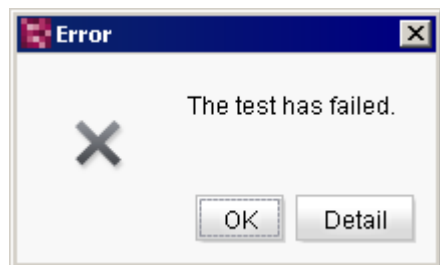


Attach date to directory name: if this option is *selected*, the current document is attached to the path specified above.

Clicking on the "Test configuration" button, which is activated by entering information in the "Path into local file system" field, allows the user to test the specified configuration. The server will first verify that all the necessary parameters have been entered. Then the server will attempt to create a folder and write a file to it in the specified "Path into local file system". It will also attempt to rename the file and then delete all of the data created. If the test is successful, the following message appears:



The following message appears if it fails:



Clicking on the "Details" button opens a dialog with the corresponding log file showing the errors.

For more information on the "Secure media" concept, see the "Documentation for the FirstSpirit security module".



7.5.10.6.2 Deployment via FTP

Deployment: FTP

Global deployment properties

Name

Deployment options

☐ Complete adjustment (for full generation only) ☒ Adjustment (without deleting)

Deployment: FTP

FTP server User

FTP server type Password

☒ Block Transfer Mode ☐ Passive Mode

Basic path on FTP server

FTP proxy settings

☐ Use FTP proxy?

Proxy server Port

☐ "USER@SITE" protocol ☒ "USER with Login" protocol

User

Password

Test configuration OK Cancel ?

Figure 7-155: Create action – Deployment via FTP

This action is used to deploy a project to a remote server via an FTP connection and therefore requires a number of settings.

General deployment properties

Name: the name of the action displayed in the schedule overview, schedule management and action templates.



Deployment options

Complete adjustment (for full generation only)

This method is used to create a data inventory identical to the generation data inventory on the web server. This means that files that are no longer present on the development server are also deleted on the web server, new files are copied, and old files already present on the server are overwritten by the new ones.

Adjustment (without deleting)

This deployment option is similar to the complete adjustment, but differs in the fact that no files are deleted from the web server, even if they are no longer present on the development server.

FTP server settings

FTP server: the address of the FTP server to which the data will be transferred. This information is mandatory.

FTP server type: the user can select the FTP server operating system in this combo box.

User: the server user who will log onto the FTP server is entered here.

Password: the password in conjunction with the user name are used to log onto the FTP server. If a user name is entered, the password must also be entered.

Block Transfer Mode: Use this checkbox to deactivate or activate the Block Transfer Mode. Deactivating this option can eliminate problems with deployment in some environments, for example ISS7. The Block Transfer Mode is activated by default.

Passive mode: if this option is *selected*, the connection to a port selected by the FTP server is established. This mode should be used when the CMS server is behind a router or a firewall is blocking the network from outside access.

Basic path on FTP server: here the user specifies the path to the directory on the FTP server where the generated data are to be transferred.

FTP proxy settings

Use FTP proxy?: if this option is *selected*, the connection to the FTP server is established via the proxy server configured in this area.

Proxy server: the address of the FTP server used to establish the connection. This information is mandatory if an FTP proxy will be used.



Port: the port to the proxy server entered above. This information is mandatory if an FTP proxy will be used.

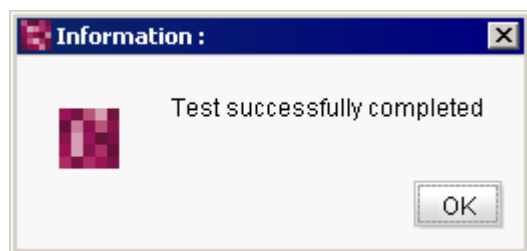
"USER@SITE" protocol: if this option is *selected*, authentication takes place for the proxy via the "USER@SITE" protocol without separate user login authentication.

"USER with Login" protocol: if this option is *selected*, proxy authentication takes place via the "User with Login" protocol with subsequent user login authentication. The user name and password are mandatory in this case.

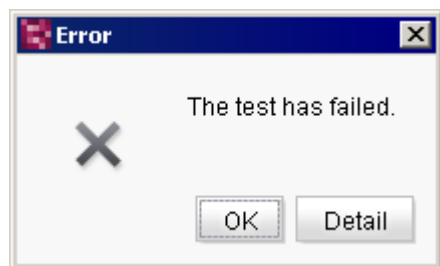
User: the user name used to establish the connection to the FTP proxy. This information is mandatory.

Password: the password along with the user name is used for FTP proxy login and is therefore also mandatory.

The specified configuration can be tested by clicking on the "Test configuration" button. The server will first verify that all the necessary parameters have been entered. The server will then attempt to connect to the FTP server and to create a folder and write a file to it in the specified "Root path on FTP server". It will then also attempt to rename the file and then delete all of the data created. If the test is successful, the following message appears:



The following message appears if it fails:



Clicking on the "Details" button opens a dialog with the corresponding log file showing the errors.



For more information on the "Secure media" concept, see the "Documentation for the FirstSpirit security module".

7.5.10.6.3 Deployment via the FirstSpirit Deployment Servlet

Deployment: FirstSpirit deployment servlet

Global deployment properties

Name:

Deployment options

☐ Complete adjustment (for full generation only) ☒ Adjustment (without deleting)

Deployment: FirstSpirit deployment servlet

Servlet URL: User:

Timeout: Seconds Password:

Path on live server:

HTTP proxy settings

☐ Use HTTP proxy?

Proxy server: Port:

User: Password:

Test configuration OK Cancel ?

Figure 7-156: Create action – Deployment via the FirstSpirit Deployment Servlet

This action is used to deploy the project via a FirstSpirit Deployment Servlet. The task of the servlet is to compare the project files on the FirstSpirit server to those on the live system. Using the CRC checksum calculation, new, modified or deleted files can be found, and only these files are updated. This differential upload accelerates the update process on the live system. (The information required for this can be read out from, among other things, an Access Control Database³⁴, which the CRC checksum manages for all objects. Use of the Access Control Database, however, is not required in order to use the CRC servlet.)

³⁴ The servlet, including the Access Control Database functions, is provided via the FirstSpirit security module and can be adapted to a specific project using the configuration dialog of the associated "FS Security WebApp" web application.



General deployment properties

Name: the name of the action displayed in the schedule overview, schedule management and action templates.

Deployment options

Complete adjustment (for full generation only)

This method is used to create a data inventory identical to the generation data inventory on the web server. This means that files that are no longer present on the development server are also deleted on the web server, new files are copied, and old files already present on the server are overwritten by the new ones.

Adjustment (without deleting)

This deployment option is similar to the complete adjustment, but differs in the fact that no files are deleted from the web server, even if they are no longer present on the development server.

Servlet settings

Servlet URL: the complete address to the FirstSpirit Deployment Servlet is entered here. This information is mandatory. After configuration of the `crcTransfer.ini` file (using FirstSpirit Security WebApp (FirstSpirit security module)), the servlet mapping is copied to the application's `web.xml` file. The servlet is mapped to `*.CRCTransfer` by default.

To test the accessibility of the servlet, the servlet can be called in the web browser, e.g. via:

`http://www.mydomain.de/fs5staging_1921116/do.CRCTransfer`

The servlet will display an error message, since no login credentials have been passed in the browser.

User: the user used by the server is entered here for the login if the servlet is used. This information is mandatory.

Password: the password is used along with the user name to log on if the servlet is used. This information is mandatory.

Timeout: if the Deployment Servlet cannot be accessed, this value specifies after how many seconds the communication attempt should be canceled.

Path on live server: here the user specifies the path to the directory on the remote server where the generated data are to be transferred. The directory specified here can potentially be deleted (under "Options – Clear generation directory beforehand" – see Chapter 7.5.10.2 page 406). The web application directory should therefore never be entered here, but rather any subdirectory can



be entered here instead.

To ensure secure access via the FirstSpirit security module (see Chapter 11.3.2 page 539), the prefix for the Access Control Database should also be entered (see Chapter 7.4.16 page 339).

HTTP proxy settings

Use HTTP proxy?: if this option is *selected*, the connection to the Deployment Servlet is established via the proxy server configured in this area.

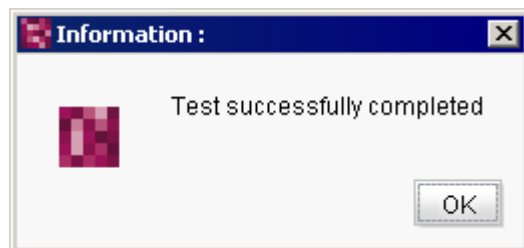
Proxy server: the address of the proxy server used to establish the connection. This information is mandatory.

Port: the port to the proxy server entered above. This information is mandatory.

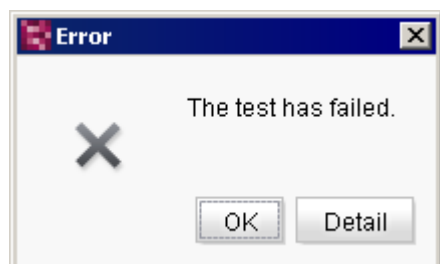
User: the user name used to establish the connection to the HTTP proxy.

Password: the password is used along with the user name to log on if an HTTP proxy is used. This information is mandatory if a user name will be used.

The specified configuration can be tested by clicking on the "Test configuration" button. The server will first verify that all the necessary parameters have been entered. The server will then attempt to connect to the Deployment Servlet and to create a folder and write a file to it in the specified "Path on live server". It will then also attempt to rename the file and then delete all of the data created. If the test is successful, the following message appears:



The following message appears if it fails:



Clicking on the "Details" button opens a dialog with the corresponding log file showing the errors.



7.6 Clustering – load distribution on generation

On the hand side, 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.

On the other hand side, FirstSpirit supports "horizontal scalability", too.

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 ServerManager.

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.



7.6.1 Concept

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:

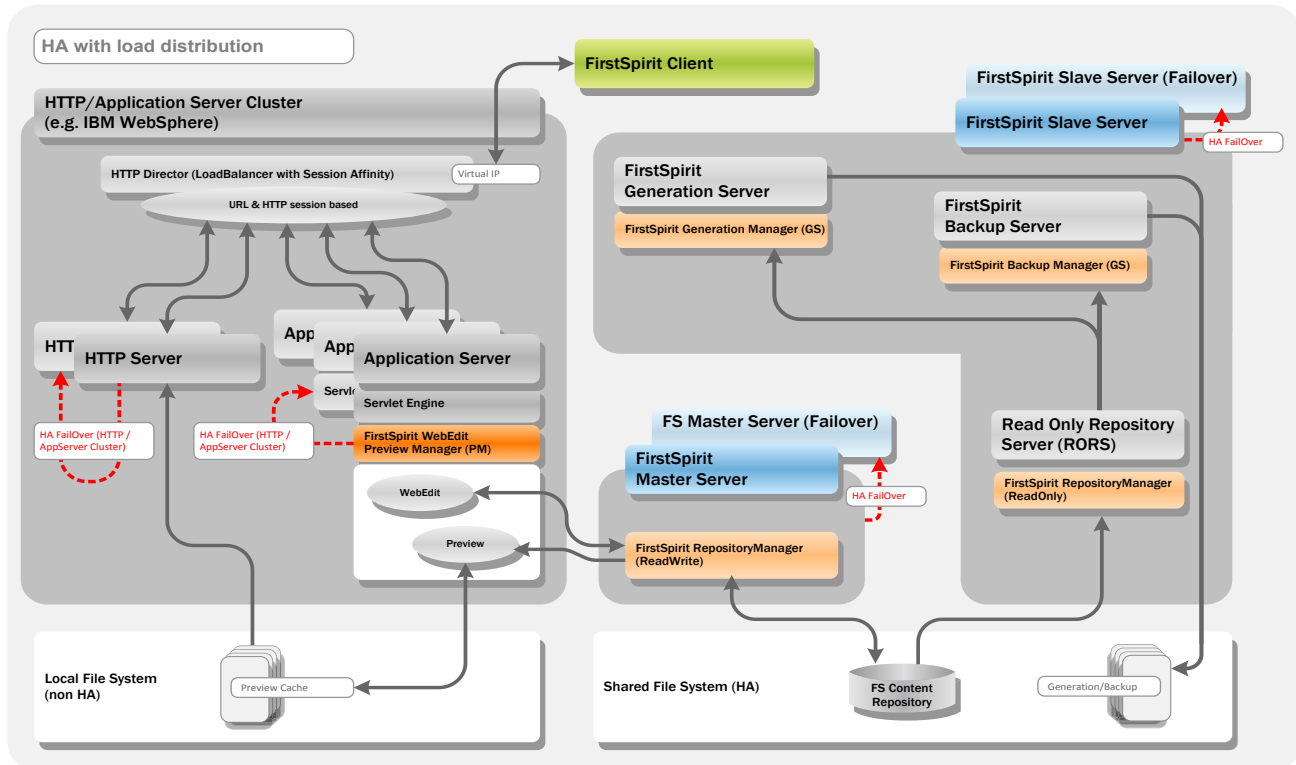


Figure 7-157: High-availability cluster with load distribution

The FirstSpirit Client (both the SiteArchitect and the ContentCreator) 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.

7.6.2 Check licence file

The "Clustering" function is a licence-dependent function. The "Clustering" menu item is only displayed in the FirstSpirit ServerManager application if a valid licence exists for this function.

Use the "FirstSpirit – Configuration – Licence" menu of the FirstSpirit ServerMonitoring to display the valid FirstSpirit functions of the `fs-license.conf` licence file (see chapter 8.6.1.2 page 471). The parameter `license.CLUSTERING` parameter must be set to the value `1` for use of the function (see Figure 7-158).

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=2226
#FIRSTspirit license
#Thu Nov 15 09:47:47 CET 2012
license.USER=e-spirit
license.EXPDATE=15.06.2013
license.MAXPROJECTS=0
license.MAXSESSIONS=0
license.MAXUSER=0
license.SOCKET_PORT=0
license.VERSION=5
license.MODULES=personalisation,search,integration,newsletter,portal,form_edit,enterprise_search
license.WEBEDIT=1
license.WORKFLOW=1
license.REMOTEPROJECT=1
license.PACKAGEPOOL=1
license.DOCUMENTGROUP=1
license.ACCESS_API=1
license.APPTAB_SLOTS=20
license.ARCHIVE=0
license.CLUSTERING=1
license.ENTERPRISE_BACKUP=1
license.HIGHAVAILABILITY=1
license.OFFICE_IMPORT=1
license.OFFICE_INTEGRATION=1
license.SCOPE=CORPORATE
license.TYPE=PRODUCTION
```

Figure 7-158: Display of the licence file parameters (ServerMonitoring)

7.6.3 Configuration of the cluster nodes

The cluster nodes are configured via the FirstSpirit ServerManager in the "Server Properties" area of the "Clustering" menu (see chapter 7.3.14 page 284).

7.6.4 Configuration of the generation schedule

A generation schedule is created and configured via the ServerManager 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 7.5.10.)

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.



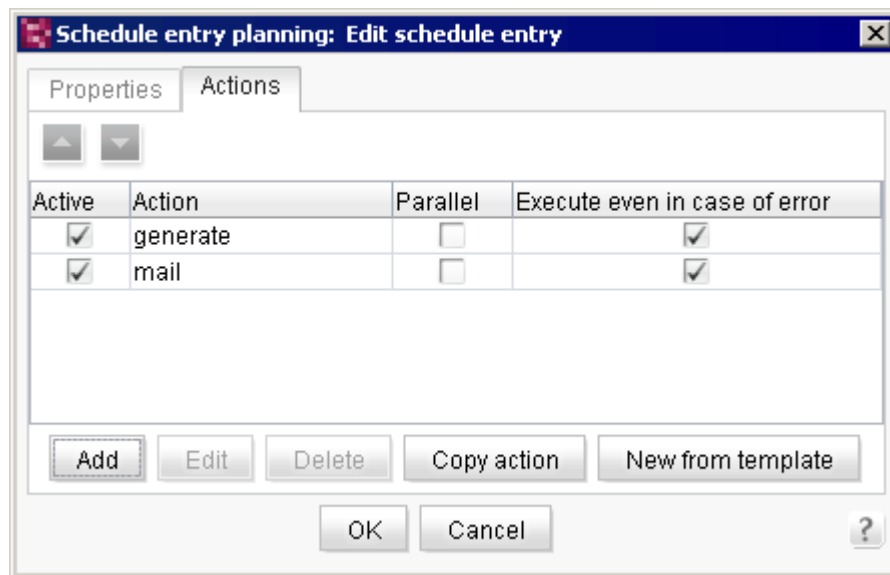


Figure 7-159: Project-related "generate" action of a generation schedule

The configuration setting for the action can be opened by clicking the "Edit" button. The following dialog opens:



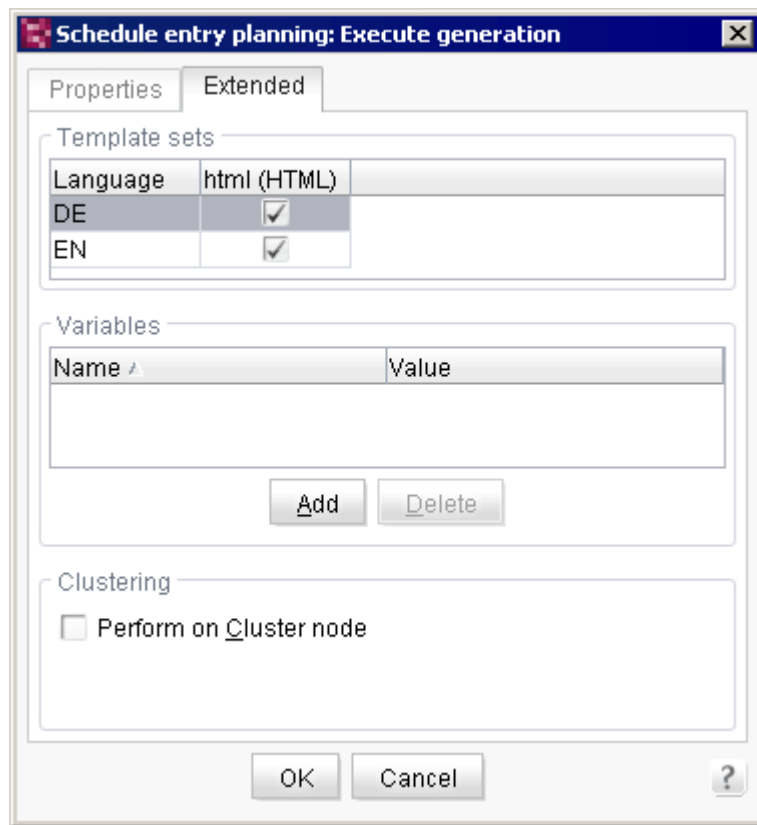


Figure 7-160: Advanced settings for the "generate" action

Apart from the known generation settings (see chapter 7.5.10), 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 ServerMonitoring (see chapter 8.6.6 page 493).

Note on using remote media for a cluster generation: Media from remote projects can be used for generations on a cluster node. The class `ClusterHelper` (package `de.espirit.firstspirit.server.clustering`) of the FirstSpirit Access-API can be used to guarantee that the remote project used in the cluster node is up-to-date. In a schedule, the cache for a project can be deleted using both `clearProjectCaches` methods in a script before cluster generation.



7.7 Configuration of the spelling check

The "SpellService" module is used for spelling checking in the FirstSpirit SiteArchitect and in the ContentCreator.

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 341). It is then possible to configure this component globally (see 7.7.3 page 431) and project-specifically (see chapter 7.7.7 page 438).
- 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 ServerManager.

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 7.3.11 Page 265).

Click the button "Install" 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:



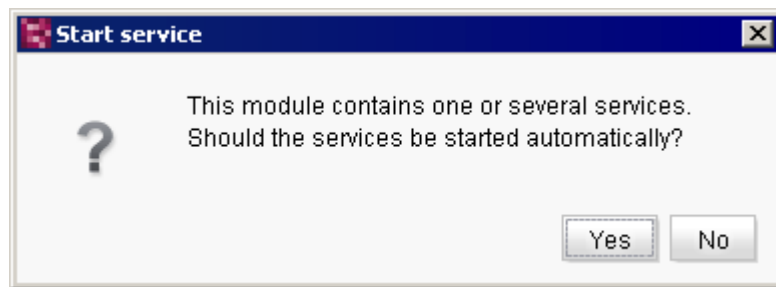


Figure 7-161: Start services automatically

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 ServerManager or via ServerMonitoring (see chapter 7.7.5 page 436).

Successful installation of the fsm file is confirmed by a message. The "SpellService" module is then listed in the "Modules" area.

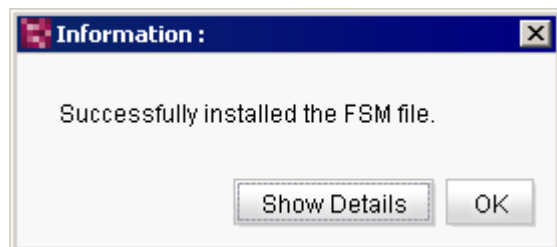


Figure 7-162: SpellService installed

Following the installation the local project component can be configured (see chapter 7.7.6 page 436).

7.7.1.2 Uninstall

The SpellService is uninstalled via the server properties in the "Modules" area (see chapter 7.3.11 page 265).

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:





Figure 7-163: 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 436). The module can only be successfully uninstalled if no more uses exist.

Click the button "Uninstall" to uninstall the module.

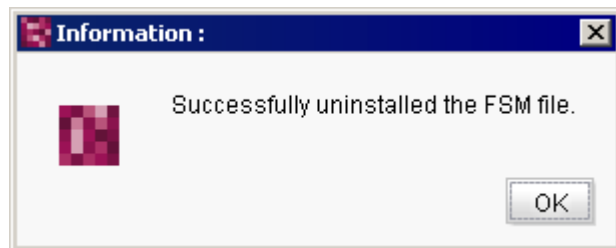


Figure 7-164: Uninstall SpellService

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.)

The update of the module takes place in analogy to the installation and is for this reason carried out by using the button "Install". The older version will be replaced by the new version of the module. The older version needs not to be uninstalled.

Following successful installation of the new version the version can also be updated within the project properties.

See also Chapter 7.3.11 page 265.



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.

Click the "Configure" 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.

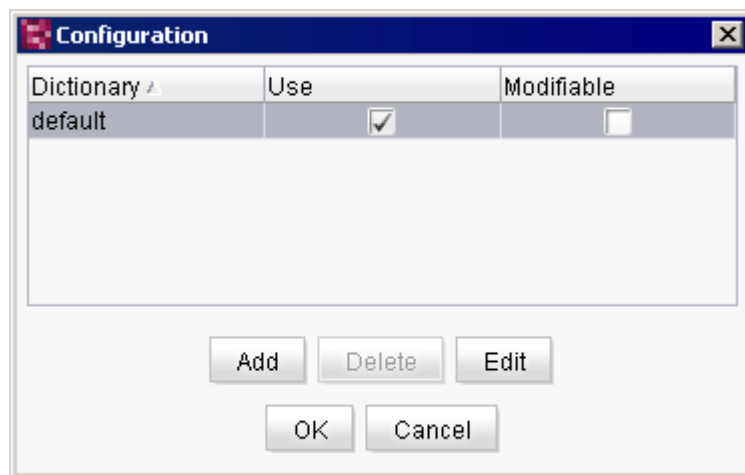


Figure 7-165: "SpellService" global configuration settings

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 "Modifiable" 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.



Modifiable: 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 "Modifiable" checkbox can only be edited if "Use" is activated.

Click the "Add" 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/modifiable) can also be adjusted in the individual project configurations.

Globally added dictionaries are filed in the server directory `conf\modules\SpellService.SpellService.`

Click the "Delete" button to remove a previously added dictionary. A confirmation prompt appears before the dictionary is deleted.

Click the "Edit" button to edit the configuration for the dictionary. The configuration dialog opens (see chapter 7.7.4 page 433).

The changes are saved by clicking the "OK" button.

The changes are not saved if the "Cancel" button is clicked; the window is closed.



7.7.4 Configure global dictionaries

Click the "Edit" button to open the dialog for configuring the dictionary.

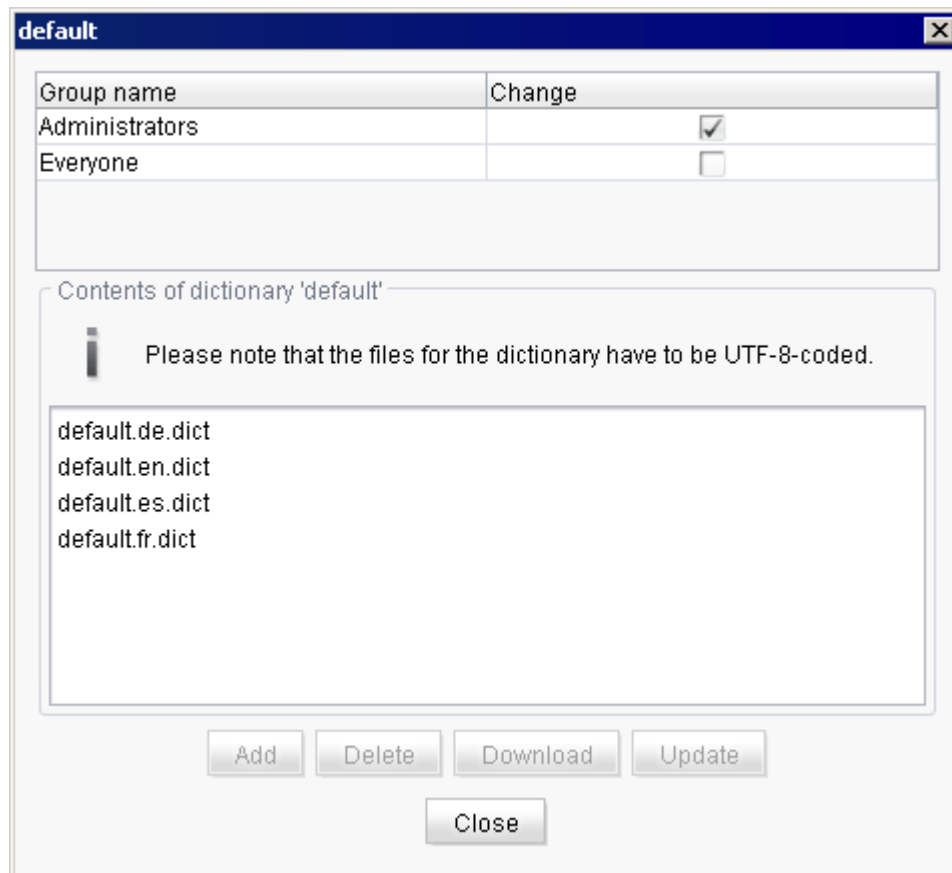


Figure 7-166: Configure dictionary

The dialog is divided into an area for rights definition (see chapter 7.7.4.1 page 433) and an area for optional addition of language-dependent contents to the dictionary (Dict files) (see chapter 7.7.4.2 page 434).

7.7.4.1 Configure permissions for global dictionaries

Group name	Change
Administrators	<input checked="" type="checkbox"/>
Everyone	<input type="checkbox"/>

Figure 7-167: 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 7-166). Other project groups are displayed in this area within the project configuration (see chapter 7.7.8 page 440).

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 440).



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

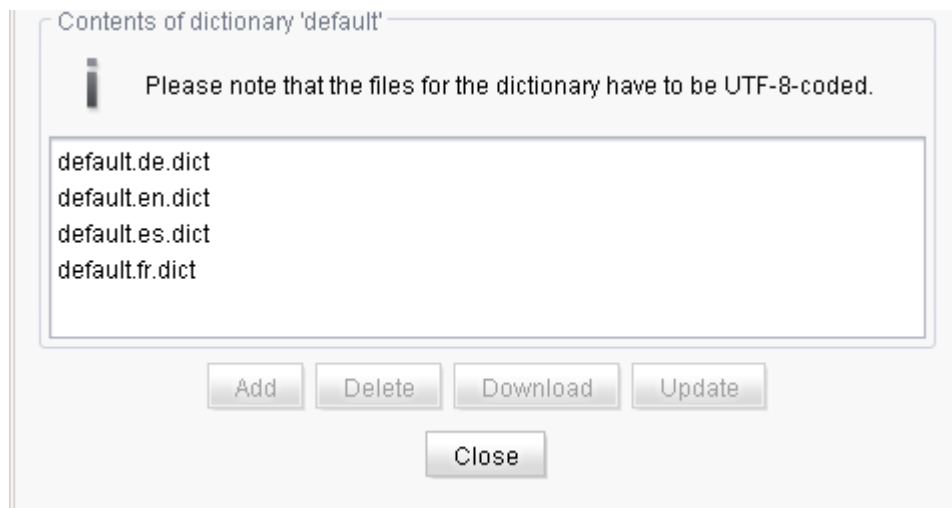


Figure 7-168: Add contents

Language-dependent contents can be added to the dictionary in the bottom part of the configuration dialog (cf. Figure 7-166). 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 SiteArchitect or ContentCreator; 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.



Click the "Add" 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:

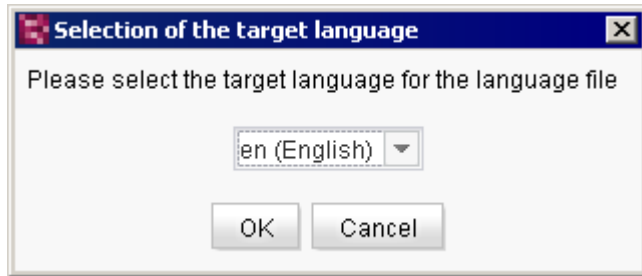


Figure 7-169: Select target language

The drop-down list displays all languages defined in the "Language Templates" area within the server properties (see chapter 7.3.6 page 257). 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



Instead of manually composed wordlists you can add freely available OpenSource dictionaries. It is important that these comply with the requirement of FirstSpirit (1 word per line).

An example are Aspell dictionaries. They can be downloaded from the official Aspell site (aspell.net). The downloaded archive must be unzipped and the ending ".dict" must be added to the included dictionary file. Then it can be added to the FirstSpirit dictionary as explained above.

Click the "Delete" button to remove a previously added word list (Dict file). A confirmation prompt appears before the dictionary is deleted.

Click the "Download" 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.



Click the "Update" 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 ServerManager within the server properties in the "Modules" area (see chapter 7.3.11 page 265)
- Via the ServerMonitoring in the FirstSpirit – Configuration – Services area (see chapter 8.6.1.7 page 475) and in the FirstSpirit – Configuration – Control area (see chapter 8.6.2.4 page 482).

7.7.6 Add SpellService as project component

Spelling checking can be activated or deactivated for a specific project. In this case the SpellService module is first installed centrally via the server properties (see chapter 7.7.1 page 428). The global configuration is also carried out via the server properties (see chapter 7.7.3 page 431). Global dictionaries can be created and configured which are then available within the SpellService project configuration (see chapter 7.7.4 page 433).

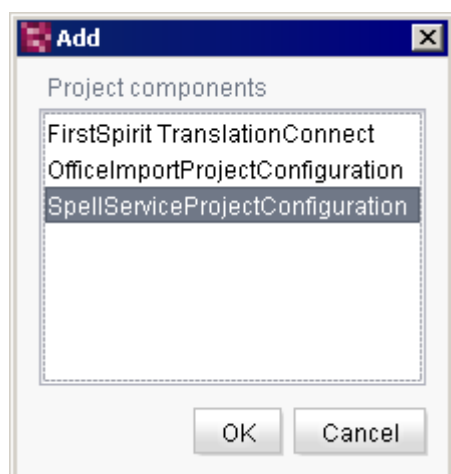
Project-specific configuration of the SpellService then takes place via the project properties in the "Project Components" area (see chapter 7.4.17 page 341).



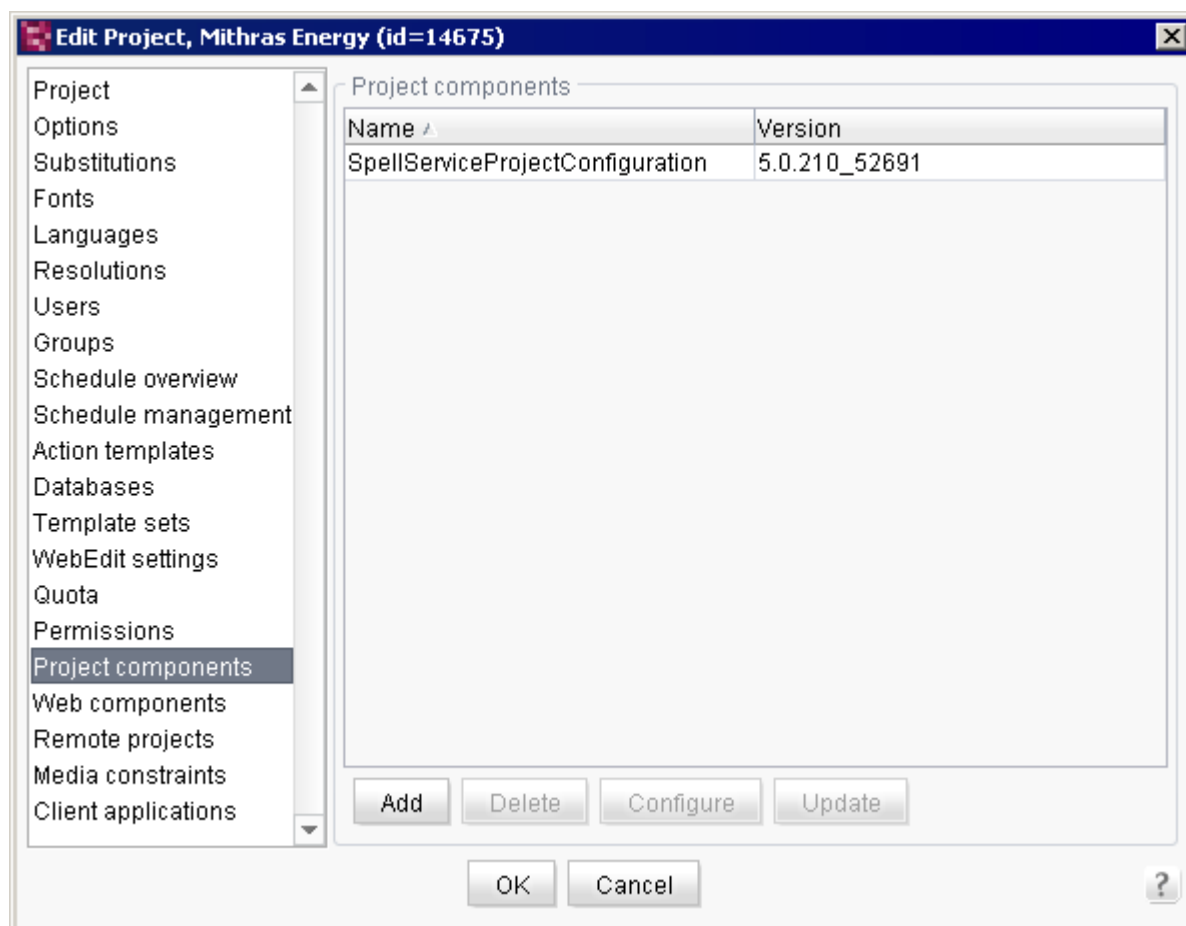
The SpellService must be started before a SpellService project configuration can take place (see Chapter 7.7.5 page 436).

Click the "Add" button to add the SpellService configuration to a project.



**Figure 7-170: Server properties**

The globally configured functions of the installed component are then available in the project.

**Figure 7-171: Project properties – SpellService Project Configuration**

Click the "Delete" button to remove a previously added SpellService Project Configuration. All project-specific SpellService configurations (project-specific dictionaries, permissions) are also removed. A confirmation prompt appears before the dictionary is deleted.

Click the "Update" button to update an existing SpellService Configuration. The button is only active if the "SpellService" module has been updated on the server (see chapter 7.7.2 page 430). In this case the version numbers of the global SpellService project configuration and the project-specific SpellService project configuration differ. The configuration settings to date (dictionaries, permissions) are retained with the update. (Depending on the new module version however, manual adjustments may be necessary.)

Click the "Configure" button to define the project-specific SpellService configuration (see chapter 7.7.7 page 438).

7.7.7 Project-specific SpellService 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.

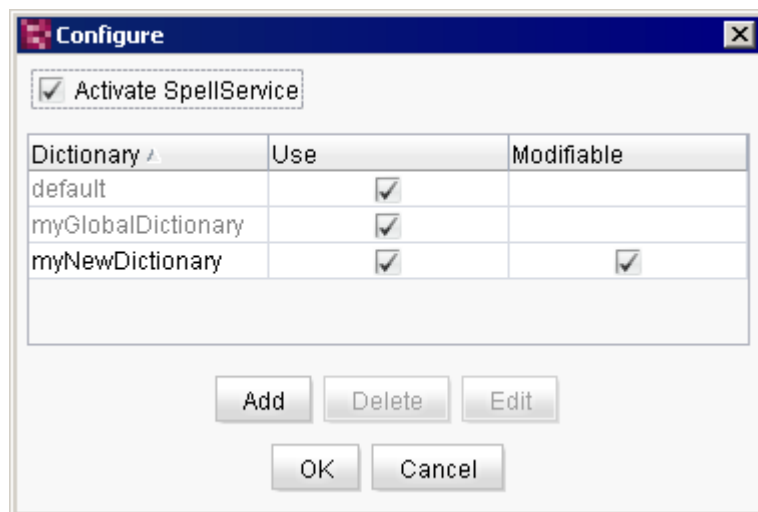


Figure 7-172: Configuration of the SpellService project configuration



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 "Modifiable" 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.

Modifiable: 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 "Modifiable" checkbox can only be edited if "Use" is activated.

If the checkbox "Activate SpellService" 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.

Click the "Add" button to add other project-specific dictionaries (see chapter 7.7.8 page 440). 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`.

A previously added project-specific dictionary can be removed with a click on the "Delete" button. 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.

Click the "Edit" button to edit the configuration for the dictionary. The dictionary configuration dialog opens (see Chapter 7.7.8 page 440).

The changes are saved by clicking the button "OK".

The changes are not saved if the "Cancel" button is clicked; the window is closed.





Importing projects: When a project with a SpellService 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 433).

Click the "Edit" button to open the dialog for configuring the dictionary (see Figure 7-166).

The dialog is divided into an area for rights definition (see chapter 7.7.8.1 page 440) and an area for optional addition of language-dependent contents to the dictionary (Dict files) (see chapter 7.7.4.2 page 434).

7.7.8.1 Configure permissions for project-specific dictionaries

Group name	Change
Administrators	<input checked="" type="checkbox"/>
Everyone	<input type="checkbox"/>
Developer	<input type="checkbox"/>
Editor	<input type="checkbox"/>
Chief Editor	<input type="checkbox"/>

Figure 7-173: 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 7-166).

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 Page 433) not only the standard groups "Everyone" and "Administrators" are available here but all groups which have access to the project (see Chapter 7.4.8 Page 315).



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 7-166). 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 434).

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 307).

7.8 Support for Apache FOP

A module provides support for Apache FOP in FirstSpirit. The module can, as usual, be installed via ServerManager (see Chapter 7.3.11 page 265):

- **fs-fop.fsm:** the "fs-fop.fsm" module is on the Apache FOP 1.00³⁵.
If error messages such as

```
Font "Symbol,normal,700" not found. Substituting with "Symbol,normal,400".
```

appear, these can be circumvented by adding the document's default font to the `font-family` parameter in `fo:root`, e.g.:

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format" font-family="Helvetica">
```

After installing the module, the affected presentation channel is available within the server properties (see also Chapter 7.3.2 page 247):

³⁵ For more information, see <http://xmlgraphics.apache.org/fop/compliance.html>



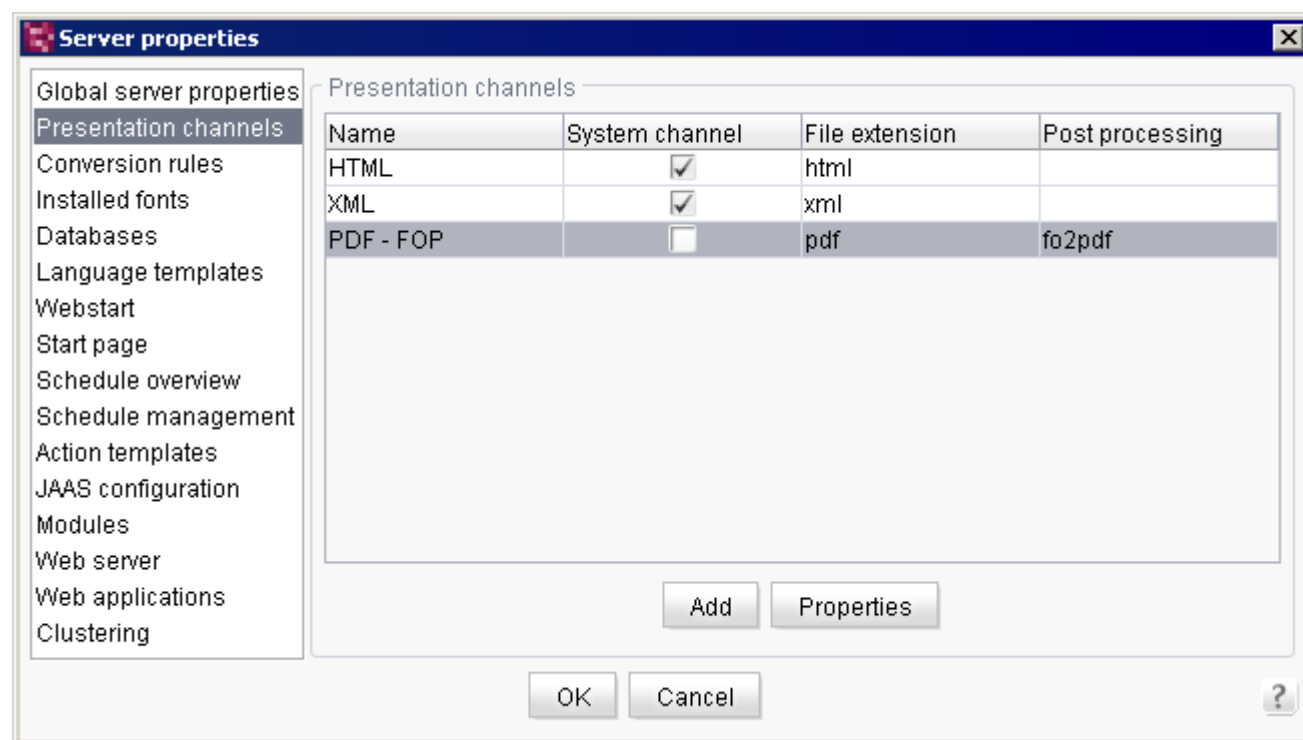


Figure 7-174: Presentation channel (server properties) after FOP installation

and can then be added to the projects as a new template set (see Chapter 7.4.13 page 326):

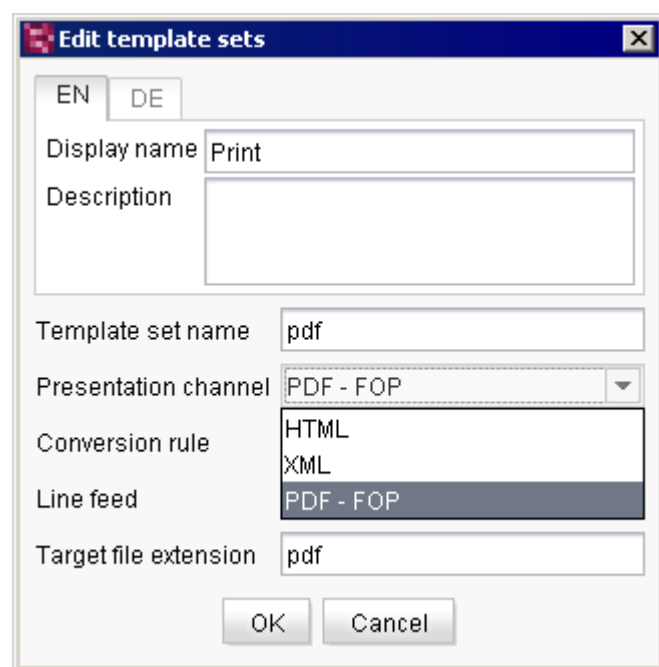


Figure 7-175: Template sets (project properties) after FOP installation



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 7.5.10.1 page 402).



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

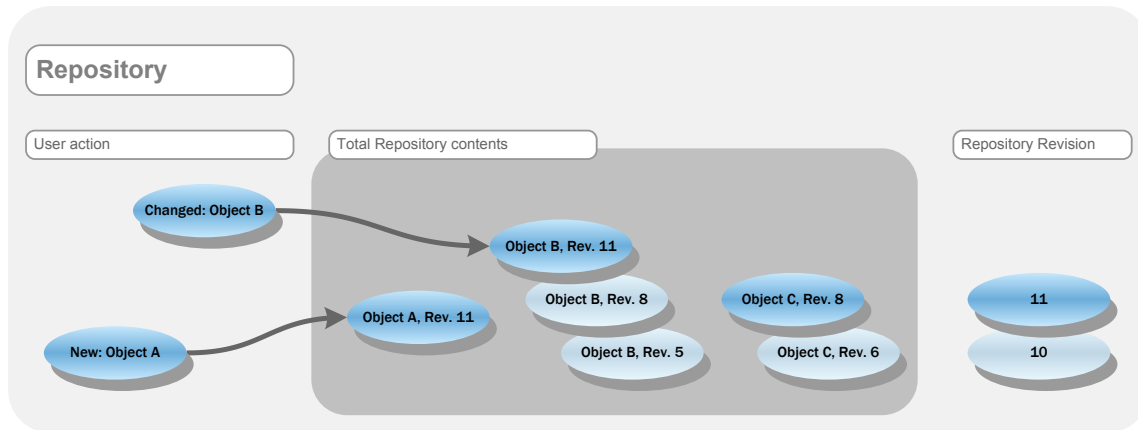
The FirstSpirit versioning and historicization concept requires that, wherever possible, all changes to objects are completely traceable in SiteArchitect and therefore earlier system states can be accessed at any time. Each time an object is changed, a new version of the object is created. This means an object has a version history which also provides information about which changes were made by which user 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 7.5.10.1 page 402) 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 7-176: 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 SiteArchitect, 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 7.5.10.1 page 402), 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 7-148 page 404), 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:

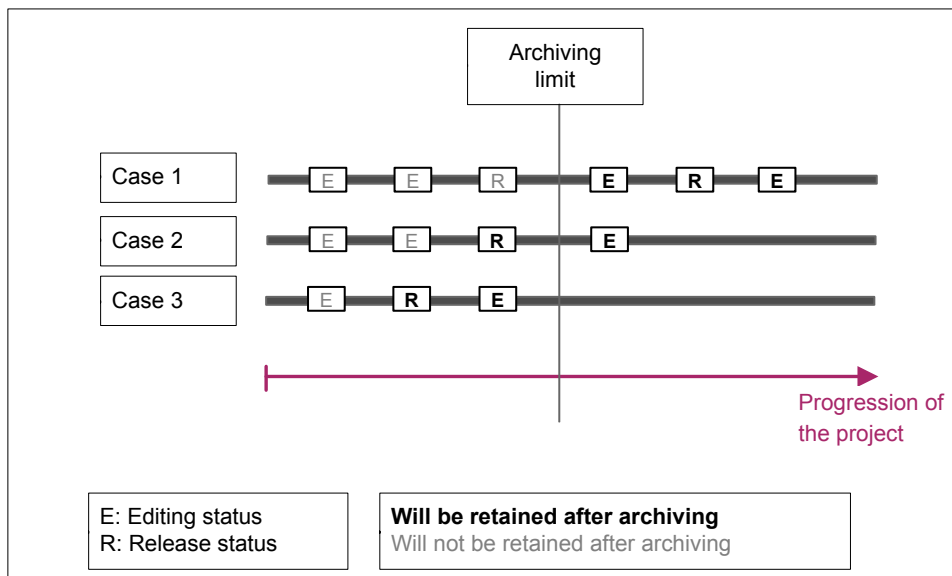


Figure 7-177: Minimum project archiving requirement

- **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 after archiving

If at the time of archival there is data (e.g. deleted objects, modification information) stored outside of the project, the particular revision in which the object was present will no longer be complete.

Revision	Date	Change on	Editor	Comment	Attributes	Child list	Content	Metadata
20518	Jun 17, 2013 2:59:18 PM	Mithras Homepage	gutknecht (Barbara Gutknecht)	Store in cache memory	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20517	Jun 17, 2013 2:58:49 PM	Mithras Homepage	gutknecht (Barbara Gutknecht)	Store in cache memory	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20139	Jun 21, 2012 11:31:06 AM	Mithras Homepage	Admin (Admin)	Workflow 'Freigabe Anfordern' Action 1...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20005	May 31, 2012 2:24:17 PM	Mithras Homepage	Admin (Admin)	Save	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16541	Sep 2, 2010 1:32:18 PM	Mithras Homepage	editor (editor)	Save	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
16516	Jun 23, 2010 1:04:05 PM	Mithras Homepage	Admin	Release by server	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Selection

1st revision: [20518] [Restore] [Display]

2nd revision (Ctrl): [] [Compare]

Options

- ☐ Show changes to ChildElements
- ☐ Show changes to Media
- ☐ Show changes to Templates
- ☐ Show hidden Revisions
- ☒ Show partially archived revisions

Figure 7-178: Version history after archiving

The object's version history then displays only the oldest **available complete** revision (which is the first revision above the archiving limit) as well as all more recent revisions (all other revisions above the archiving limit, 1. in Figure 7-178). These are displayed in bold in the version history. They can be displayed ("Display" button), included for comparisons with other complete revisions ("Compare" button) and restored ("Restore" button).

If there are no complete revisions present (e.g. if the "Do not maintain version history (complete archiving)" option was selected; see Figure 7-147), the **last released status** (2. in Figure 7-178, in this example, it is below the archiving limit) and, if present, the **most recent editing status** will always be displayed (also refer to section 7.9.3, page 444 for more information). They can be included for comparisons with other complete revisions ("Compare" button) and restored ("Restore" button, also refer to section 7.9.4.1, page 447 for more information). It may also no longer be possible to display the information correctly ("Display" button), since the required revisions of underlying templates may no longer be present for display after performing an archival. The following message will appear Figure 7-181.



Revisions that are **no longer present in their entirety** can be displayed using the "Show partially archived revisions" option (3. in Figure 7-178). They can be displayed ("Display" button) only and may not be historically accurate; see the message in Figure 7-178.

7.9.4.1 Restore following archiving (function "Restore")

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:

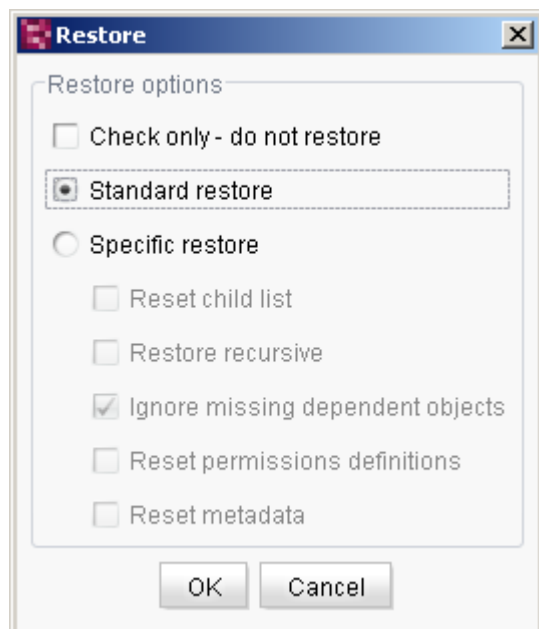


Figure 7-179: Restoration following archiving

7.9.4.2 Preview of archived revisions (function "Display")

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:



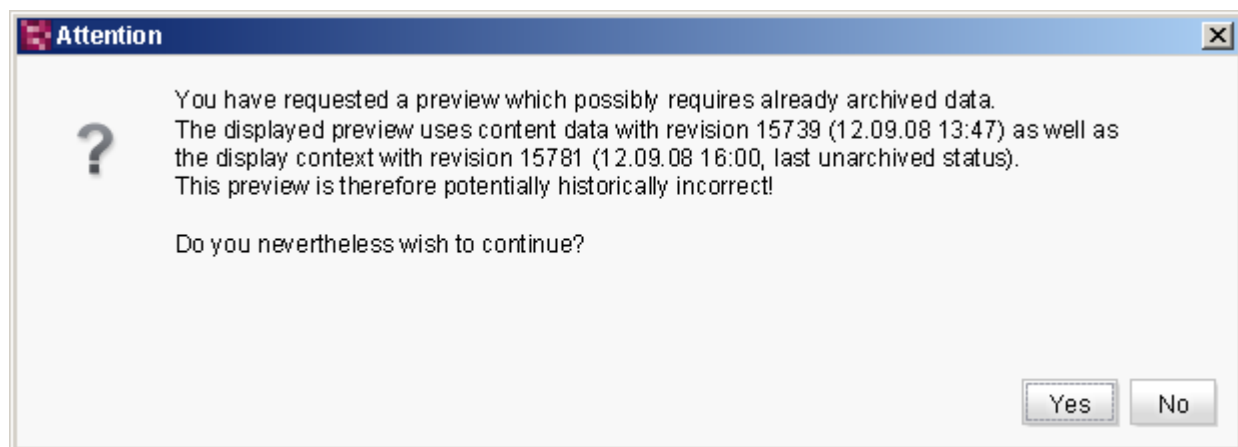


Figure 7-180: Message issued for historically incorrect preview

7.9.4.3 Comparison of revisions following archiving (Function "Compare")

The following message is displayed if all parts of the objects necessary for a comparison are no longer available following archiving:



Figure 7-181: Message issued for version comparison following archiving



8 FirstSpirit ServerMonitoring

The browser-based FirstSpirit ServerMonitoring is a web application used to monitor the FirstSpirit server. This application is used to display the current operational parameters such as memory utilization, number of users and much more. FirstSpirit ServerMonitoring is used similarly to ContentCreator through a web browser.

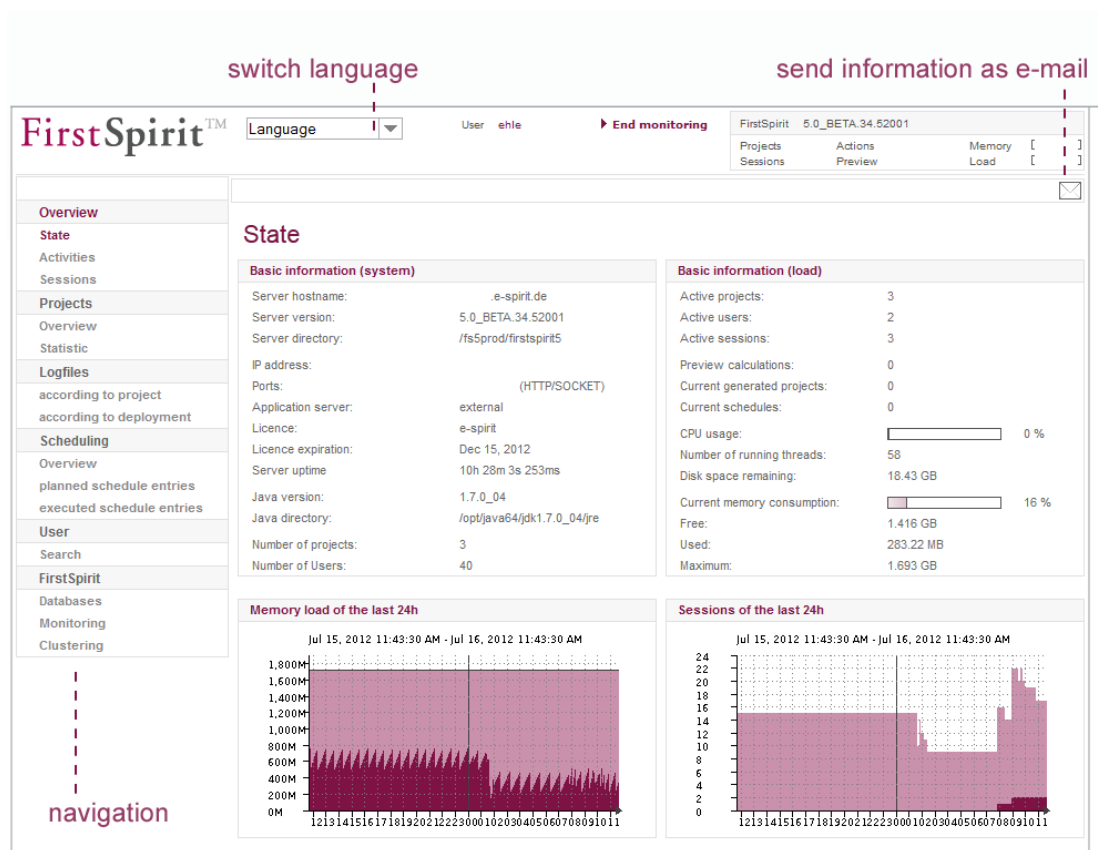


Figure 8-1: FirstSpirit ServerMonitoring – Overview status

On the left-hand side of the screen is the FirstSpirit ServerMonitoring navigation area. Some menu levels have additional submenus that can be displayed in another navigation area at the top of the page.

Current server status information can also be sent via (text) e-mail.





Logging onto multiple FirstSpirit servers simultaneously with the same host names (e.g. myServer:8200 and myServer:8400) via a web browser is not supported.



The complete range of functions available to server administrators is documented [here](#). Project administrators also have access to FirstSpirit ServerMonitoring, but not all menu levels are available to them.

8.1 Overview

8.1.1 Overview – Status

This page provides an overview of the most important information (see Figure 8-1). The page is divided into four areas:

Basic information (system): this area shows the general server configuration information.

Basic information (load): this area shows general information on the current server load.

Memory load of the last 24h: this area provides an overview of the server's memory load over the past 24 hours. The black line indicates the defined -Xmx value from the configuration file `fs-wrapper.conf` (see Chapter 4.3.2.3, `wrapper.java.maxmemory` parameter). The light purple section is the area reserved in the operating system by Java VM for the Java heap. The dark purple section is the actual heap used. Clicking on the graph opens the "FirstSpirit – Monitoring – Memory" area (see Chapter 8.6.5.1, starting on page 486).

Sessions of the last 24h: this area provides an overview of the number of sessions within the past 24 hours. Clicking on the graph opens the "FirstSpirit – Monitoring – Sessions" section (see Chapter 8.6.5.2, starting on page 487).




8.1.2 Overview – Activities

This page provides an overview of some server activities:

Activities

Current schedules:

Mithras_12	Rebuild search index	Stop execution
 Name:	Rebuild search index (?)	Start date:
Status:	Running	Runtime to date:
		16.07.2012 12:02:40
		4s 43ms

Planned schedules

Project-based

Server-wide

Dokumentation WebClient (PRODUKTIV) - Generate PDF Document:

Start date: 16.07.2012 18:00:00

- Generate
- Mail

Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV) - generate (daily):

Start date: 16.07.2012 18:00:00

- Generate
- Mail

Dokumentation WebClient (PRODUKTIV) - generate (daily):

Start date: 16.07.2012 18:00:00

- Generate
- Mail

Scheduling of the last 24 hours (max. 15 entries)

Dokumentation WebClient (PRODUKTIV)

Name	Status	Start	End
Generate PDF Document	Generate [Successful] Mail [Successful]	16.07.2012 01:54:39	16.07.2012 01:55:57
Generate WebEdit Tooltips	Generate [Error] Mail [Successful]	16.07.2012 01:54:39	16.07.2012 01:55:13
generate full	Generate [Successful] Mail [Successful]	16.07.2012 01:54:39	16.07.2012 01:55:45

Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV)

Name	Status	Start	End
generate full	Generate [Successful] Mail [Successful]	16.07.2012 01:54:39	16.07.2012 02:01:18
build dvd-index	Generate [Successful] Mail [Successful]	16.07.2012 01:54:39	16.07.2012 01:55:14

Active projects

Project	Sessions	User	Reference calculation	Search indexing
<u>Dokumentation WebClient (PRODUKTIV)</u>	1	1	Not started	Not started
<u>Online-Dokumentation FirstSpirit...</u>	1	1	Not started	Not started

Figure 8-2: FirstSpirit ServerMonitoring – Overview – Activities

Current schedules: this table lists all schedule entries that are currently active on the server (see Chapter 7.5 page 370).



Planned schedules: this table lists all upcoming, active (not manual) schedule entries (see Chapter 7.5.1.2 page 373).

Scheduling of the last 24 hours (max. 15 entries): this table lists all schedule entries executed within the past 24 hours.

Active projects: this table lists all projects that are currently active on the server. The following information is displayed:

While a schedule entry is running, additional **schedule-specific information** is displayed. During generation, information such as the start time, the progress of the schedule entry so far and the estimated remaining runtime as well as the average duration of the generation per page is displayed. In addition, the overview shows how many pages have been generated up to a certain point during project generation.

Clicking on an active project displays a more detailed overview of this project (see Chapter 8.2.1, starting on page 454).

Sessions: number of currently active sessions in the project.

User: number of users currently logged in to the project.

Reference calculation: status of the reference calculation in the project (see Chapter 9.15).

Search indexing: status of search indexing (see Chapter 9.18).

8.1.3 Overview – Sessions

This page includes a table listing all of the currently running sessions.

Session ID: unique ID of the session.

Type: type of session. A distinction is made between the following session types:

- WEB: session is established through the start page.
- ContentCreator: session is established through ContentCreator.
- Main: session is established through SiteArchitect.
- Child: subordinate, internal session that is established, for instance, when editing a project in SiteArchitect or through ServerManager. *Note:* when editing a project in ServerManager, a CHILD session is opened for each project being edited and it is closed again after the Java application is closed.
- Web monitor: session is established through ServerMonitoring.



- **Dummy:** internal sessions that are established when generation is executed, for instance. These sessions are only displayed when the box **"Show all sessions"** has been checked. These sessions are not included in the evaluation of licensed sessions (see Chapter 4.3.5 page 101).
- **Remote:** session is established through a remote project.
- **Staging:** session is established when generating a preview of the generated project status (via the URL specified for the generation task). These sessions are not included in the evaluation of licensed sessions (see Chapter 4.3.5 page 101).



In the case of a child type session, it is possible to send the session user a message. Clicking on the relevant session ID opens the page "FirstSpirit – Message" (see Chapter 8.6.3, starting on page 485). In this case, however, the message is sent only to the user of the selected session and not to all users on the server.

User: user login name. Clicking on the user of a session brings up detailed information linked to this user on the page "Users – Search" (see Chapter 8.5.1, starting on page 469).

Project: project in which the particular user is working. Clicking on the project of a session brings up detailed information linked to this project on the page "Projects – Overview" (see Chapter 8.2.1.1, starting on page 454).

The **"End"** link provides the ability to terminate a session without warning users in advance. It is important to note that all unsaved entries made during the particular session will be lost.

Access: indicates the point in time at which the particular user last accessed the server.

Login time: indicates the point in time at which the particular user logged onto the server.

This and other information can also be requested via JMX monitoring (see Chapter 9.22 page 520).




8.2 Projects

8.2.1 Projects – Overview

This page includes a table of all projects installed on the server. In addition to the project name and description, the table lists the number of users permitted to access the individual project and whether the project is activated or deactivated on the server. The current schedule status of the project and the project ID are also displayed.

This and other information can also be requested via JMX monitoring (see Chapter 9.14 page 512).

This list can be sorted by project name, user, activation status or project ID. The option "**Active projects only**" is used to hide all deactivated projects. Clicking on the  button updates the project overview.



Project administrators see only the projects for which they are registered as administrator.

Clicking on a project in the list displays detailed information on that project.

8.2.1.1 Project details

The project details include:

Project name, description as they are defined in the project properties.

Project ID unique project ID for the server.

Activated indicates whether the project is activated or not.

Max. sessions indicates the maximum number of sessions.

Last accessed, last changed, last release, statistics start time shows the point in time at which these actions were carried out.

Number of users shows the number of users registered for this project.



Users shows the logins of the users registered for this project.

Last used shows the last 5 users of the project.

There is also a table listing all groups defined for the project and their associated users.

Show import log clicking on this button displays the project import log file.

Show import ID map: clicking on this button displays the project import log map. When importing a project, the "old" IDs are replaced by the new values. The "old" IDs replaced with "new" ones are shown in the ID map.

8.2.2 Projects – Statistics

This page is split up further into different areas.

- **Accesses:** access statistics (see Chapter 8.2.2.1, starting on page 455)
- **Deployment:** deployment statistics (see Chapter 8.2.2.2, starting on page 456)
- **Resources:** resource statistics (see Chapter 8.2.2.3, starting on page 457)

8.2.2.1 Accesses

Information on accesses to the installed projects is available on this page. A table lists all of the projects installed on the server. In addition to the project name, information is provided on the point in time at which the project was last accessed and last changed. The table also shows which user last accessed the project, how many users are permitted to access the project and the maximum number of sessions that were active at the same time.

This list can be sorted by project name, user, last access, last change or maximum number of sessions. The **"Active projects only"** option is used to hide all deactivated projects.



Project administrators see only the projects for which they are registered as administrator.

Clicking on a project in the list displays detailed information on that project. (See Chapter 8.2.1.1, starting on page 454.)



8.2.2.2 Deployment

Information on the deployments of projects is available on this page. The table lists all of the projects installed on the server. Below the projects are schedule entries that have been executed for the projects. In addition to the schedule entry name, the table provides information on the number of executions that have taken place so far, the duration of the last execution and the average duration of a schedule entry execution.

This list can be sorted by any table column. The option **"Active projects only"** is used to hide all deactivated projects.

Clicking on a listed schedule entry, e.g. a deployment, displays detailed information about the schedule entry execution in this project:

Deployment

Schedule: generate (daily)

Project: Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV)

ID	158611
Name	generate (daily)
Description	
Type	daily
Status	active
Project name	Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV)

Number of executions	6
Last duration	4m 27s 280ms
Average duration	3m 0s 144ms
last execution	16.07.2012 18:00:00 – 16.07.2012 18:04:27

Name	Project name	Start	Status	Error	User	Duration	
generate (daily)	Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV)	16.07.2012 18:00:00	[Successful]	0F / 0E / 0W	SYSTEM	4m 27s 280ms	Show history
» Generate		16.07.2012 18:00:00	[Successful]	0F / 0E / 0W		4m 27s 215ms	Show history
» Mail		16.07.2012 18:04:27	[Successful]	0F / 0E / 0W		56ms	Show history
generate (daily)	Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV)	13.07.2012 18:00:00	[Successful]	0F / 0E / 0W	SYSTEM	2m 13s 942ms	Show history
» Generate		13.07.2012 18:00:00	[Successful]	0F / 0E / 0W		2m 13s 882ms	Show history
» Mail		13.07.2012 18:02:13	[Successful]	0F / 0E / 0W		55ms	Show history

Figure 8-3: FirstSpirit ServerMonitoring - Deployments for a project

This view contains additional schedule-specific information and shows, for instance, how many deployments have been performed for this project so far and the size of the average or last duration of the generations so that it is possible to estimate whether changes recently made have an effect on the project's performance.





Project administrators see only the projects for which they are registered as administrator.

8.2.2.3 Resources

Information on the resources required for the projects is available on this page. The amount of disk space required for each project is listed here.

The option "**Active projects only**" is used to hide all deactivated projects.



Project administrators see only the projects for which they are registered as administrator.

Clicking on a project in the list displays detailed information on that project. (See Chapter 8.2.1.1, starting on page 454.)



8.3 Log files


8.3.1 Log files – complete server



This menu level is only available to server administrators.

This view is divided into the following areas:

- Current view (see Chapter 8.3.1.1 page 459)
- History (see Chapter 8.3.1.2 page 460)
- Search (see Chapter 8.3.1.3 page 461)

 This icon is used to download log files within the application for ServerMonitoring and/or to send as an e-mail attachment. When opening the mail transfer dialog using this icon, the relevant log files are attached automatically:

Mail transfer

Recipient



helpdesk@e-spirit.de ▼

Re:

WEBMonitor: Protocol file

Comment

Attachment

fs-server INFO 20120717104557_20120717104502.log (1,387 KB)

Submit

Cancel

Figure 8-4: Mail transfer with attached log file



The user can also download and save the files within the dialog by clicking on the link.

8.3.1.1 Current view

The most recent log file actions are output on this page.

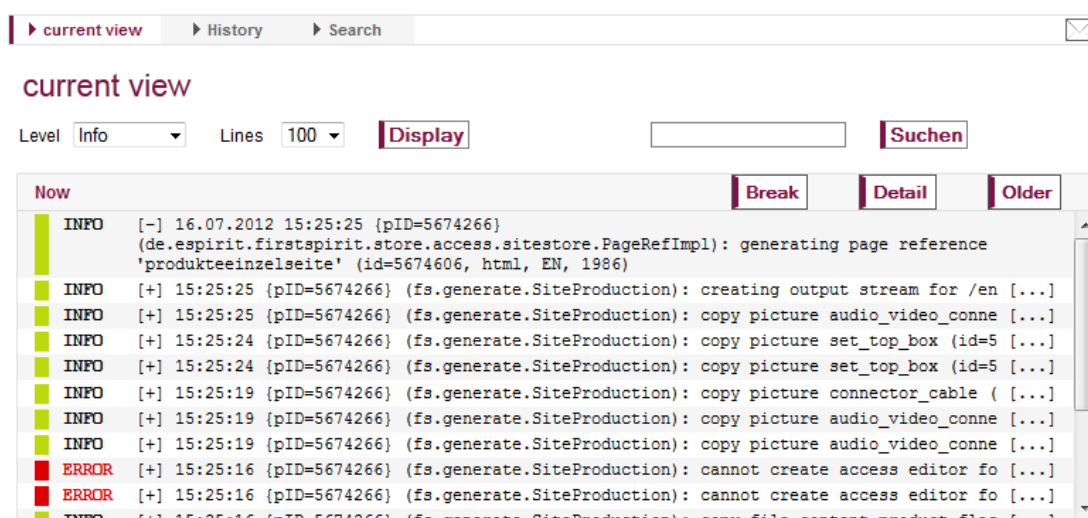


Figure 8-5: Log files – Current view

The heading lines contain information on the period within which the server actions took place.

"Now" means that the list is updated continuously and the most recent actions are displayed at the top of the list.

Clicking on the **Break** button pauses updating of the list so that the user has time to view some entries without rushing. Clicking on the plus sign [+] next to each action displays the entry in its entirety.

Clicking on the **Detail** button allows the user to view all entries at once in their entirety.


Clicking on the **Older** button allows the user to output an older section of the log file. (See Chapter 8.3.1.2, starting on page 460.)

Above the log list are some filter options for the current view:

Level: here the user can set the particular information level for the display of server actions. The levels available to choose from are Debug, Info, Warning and Error.



Lines: here the user can select how many lines from the server log are to be displayed simultaneously.

Clicking on the  button refreshes the view.

Find The search function allows the user to search for particular text fragments in the log file. The search is case sensitive. The search results are displayed below the current log list.

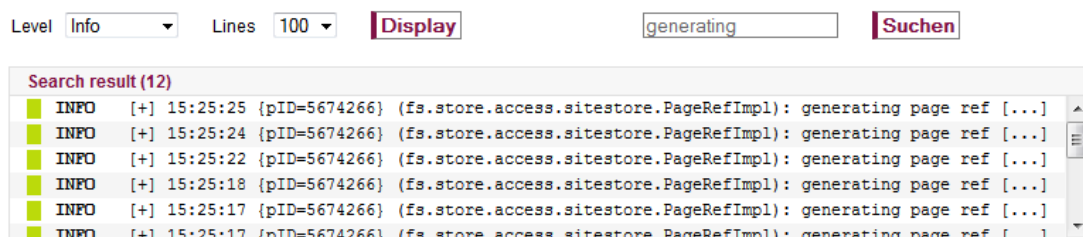




Figure 8-6: Log files – Current view – Search results


8.3.1.2 History

This page allows the user to output an older section of the log file. The view and filter options (level, lines) for the most part correspond to "Current view" in Chapter 8.3.1.1.

More input options are provided here with the addition of a date and time selection:

Date: clicking on the  button opens a window where a date can be selected. The arrow buttons next to the name of the month can be used to navigate to the next or previous month. Clicking on the desired day closes the window and places the selected day in the field.

Time: clicking on the  button opens a window where a time can be selected (clicking on it again will close the window). Clicking on the desired time closes the window and places the selected time in the field.

Clicking on the  button refreshes the view.

The heading lines contain information on the period within which the server actions took place. Clicking on the plus sign [+] next to each action displays the entry in its entirety.

Clicking on the  button allows the user to view all entries at once in their entirety.



Find The search function allows the user to search for particular text fragments in the log file. The search is case sensitive. The search results are displayed below the current log list.

8.3.1.3 Search

This page allows the user to output a certain period of time from the log file. The output is restricted using search criteria.

The screenshot shows the 'Search' tab in the FirstSpirit interface. At the top, there are tabs for 'current view', 'History', and 'Search'. Below the tabs, the search criteria are set to 'Period' from '16.07.2012 11:32:21' to '17.07.2012 10:05:41'. The search options are set to 'Search word' with the value 'generate'. The level is set to 'Info' and hits per page are set to '10'. A 'Find' button is visible. Below the search criteria, there is a table of search results. The table has columns for 'Log extract', 'Lines before', and 'Lines after'. The results show log entries for 'fs.generate.SiteProduction' and 'fs.store.access.sitestore.PageRefImpl'.

Log extract	Lines before	Lines after
INFO [-] 17.07.2012 10:05:41 {pID=6045153} (de.espirit.firstspirit.generate.SiteProduction): creating output stream for /Header navigation/Site map/index.xml	3	3
INFO [+] 17.07.2012 10:05:41 {pID=6045153} (fs.generate.SiteProduction): creating output str [...]	3	3
INFO [+] 17.07.2012 10:05:41 {pID=6045153} (fs.generate.SiteProduction): creating output str [...]	3	3
INFO [+] 17.07.2012 10:05:41 {pID=6045153} (fs.generate.SiteProduction): creating output str [...]	3	3
INFO [+] 17.07.2012 10:05:41 {pID=6045153} (fs.generate.SiteProduction): creating output str [...]	3	3
INFO [+] 17.07.2012 10:05:41 {pID=6045153} (fs.generate.SiteProduction): creating output str [...]	3	3
INFO [+] 17.07.2012 10:05:41 {pID=6045153} (fs.generate.SiteProduction): creating output str [...]	3	3
INFO [+] 17.07.2012 10:05:41 {pID=6045153} (fs.generate.SiteProduction): creating output str [...]	3	3
INFO [+] 17.07.2012 10:05:41 {pID=6045153} (fs.generate.SiteProduction): creating output str [...]	3	3
INFO [+] 10:05:42 {pID=6045153} (fs.store.access.sitestore.PageRefImpl): generating page ref [...]	3	3
INFO [+] 10:05:42 {pID=6045153} (fs.generate.SiteProduction): creating output stream for /He [...]	3	3
INFO [+] 10:05:42 {pID=6045153} (fs.store.access.sitestore.PageRefImpl): generating page ref [...]	3	3
INFO [+] 10:05:41 {pID=6045153} (fs.generate.SiteProduction): creating output stream for /He [...]	3	3
INFO [+] 10:05:41 {pID=6045153} (fs.store.access.sitestore.PageRefImpl): generating page ref [...]	3	3
INFO [+] 10:05:41 {pID=6045153} (fs.generate.SiteProduction): creating output stream for /He [...]	3	3
INFO [+] 10:05:41 {pID=6045153} (fs.store.access.sitestore.PageRefImpl): generating page ref [...]	3	3

Figure 8-7: Log files – Search

Period: the two date and time selection fields can be used to specify the period for which the log file actions are to be output. Clicking on the button opens a window where a date can be selected; clicking on the button opens a window where a time can be selected. A one day search interval is always set by default.




Search options: if the **Search word** option is selected, the user can search for particular text fragments in the log file. If the **Exceptions** option is selected, all actions within the specified period are output.

Level: here the user can set the particular information level for the display of server actions. The levels available to choose from are Debug, Info, Warning and Error.


Hits per page: here the user can select how many lines from the server log are to be displayed simultaneously.

Clicking on the  button refreshes the view.

The heading lines contain information on the period within which the server actions took place. Clicking on the plus sign [+] next to each action displays the entry in its entirety.

Clicking on the  button allows the user to view all entries at once in their entirety.

Clicking on a log entry in the list also populates the **Log extract** section, which displays the log entries before and after the selected entry (number can be defined using "Lines before" / "Lines after") (see Figure 8-7).

 Clicking on the button displays the selected entry in the associated log file. The view in service monitoring switches to the "Log files / History" area (see Chapter 8.3.1.2 page 460).

8.3.2 Log files – by project



Project administrators see only the projects for which they are registered as administrator.

This page includes a table of all projects installed on the server. The project name and description are displayed here.


Clicking on the Show log file link next to the project entry displays the selected project's current log view.



Clicking on the [Project details](#) link lets the user view additional information on the selected project (see Chapter 8.2.1.1, starting on page 454).

This view is divided into the following areas:

- Current view
- History
- Search

 This icon is used to download log files within the application for ServerMonitoring and/or to send as an e-mail attachment (for more information, see Chapter 8.3.1).

8.3.2.1 Current view

The most recent log file actions for the selected project are output on this page. The log list is used to specify the name of the selected project. The [Change](#) link next to the project name is used to select a different project for viewing.

Detailed documentation on the layout of the current log view can be found in Chapter 8.3.1.1, starting on page 459.

8.3.2.2 History

This page allows the user to output an older section of the log file for the selected project. The [Change](#) link next to the project name is used to select a different project for viewing.

Detailed documentation on the layout of the history log view can be found in Chapter 8.3.1.2, starting on page 460.

8.3.2.3 Search

This page allows the user to output a certain period of time from the log file for the selected project. The [Change](#) link next to the project name is used to select a different project for viewing.

Detailed documentation on the search for particular log file actions can be found in Chapter 8.3.1.3, starting on page 461.



8.3.3 Log files – by deployment



Project administrators see only the projects for which they are registered as administrator.

Clicking on the project name displays detailed information on that project. (See Chapter 8.2.1.1, starting on page 454.)

This page includes a table of all projects installed on the server. The project name and description are displayed here.

Clicking on the [Show log file](#) link next to the project entry displays the selected project's current log view.

Clicking on the [Project details](#) link lets the user view additional information about the selected project (see Chapter 8.2.1.1, starting on page 454).

This view is divided into the following areas:

- Current view
- History



This icon is used to download log files within the application for ServerMonitoring and/or to send as an e-mail attachment (for more information, see Chapter 8.3.1).

8.3.3.1 Current view

The log file of the selected project's currently running deployment is output on this page. The log list is used to specify the name of the selected project. The [Change](#) link next to the project name is used to select a different project for viewing.

Detailed documentation on the layout of the current log view can be found in Chapter 8.3.1.1, starting on page 459.

8.3.3.2 History

This page allows the user to output the log file of an older deployment. A list of all project installed on the server is displayed.



The user can specify above the project list the period during which the deployments to be found are to have taken place. The user can also specify the maximum number of deployments to be listed for each project. Clicking on the **Find** button updates the project list.

The screenshot shows the 'History' tab selected. At the top, there are tabs for 'current view' and 'History'. Below the tabs, the 'History' section is titled. A search bar is present with a 'Find' button. The search criteria are set to 'Period' with a 'From' date of '16.07.2012' and a 'To' date of '14.05.2012'. The 'Max. Entries per project' is set to '100'. A 'Search' button is located to the right of the search criteria. Below the search bar, the results are displayed as a list of deployment entries. Each entry includes a plus sign (+) to expand the view, the project name, the number of log files, the time range, the status (e.g., 'Successful'), and links for 'Show history' and 'Download'. The entries are listed in descending order of time.

History

Period From 16.07.2012 To 14.05.2012 Max. 100 Entries per project Search

(QA) FirstUnit 5.x (PRODUKTIV) - [No protocols exist in the period indicated]

[+] Dokumentation WebClient (PRODUKTIV) (100) - [16.07.2012 18:00:00 - 22.05.2012 09:57:22]

[+] Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV) (48) - [16.07.2012 18:00:00 - 14.05.2012 09:23:08]

Show more recent entries - Show older entries

Monday, 16.07.2012

[+] build dvd-index (2) - [18:00:00 - 18:00:29] - [Successful] (0F / 0E / 0W) - Show history - Download (1.692 KB)

[+] generate (daily) (2) - [18:00:00 - 18:04:27] - [Successful] (0F / 0E / 0W) - Show history - Download (1.698 KB)

[+] generate full (2) - [01:54:39 - 02:01:18] - [Successful] (0F / 0E / 0W) - Show history - Download (1.679 KB)

- Generate - [01:54:39 - 02:01:18] - [Successful] (0F / 0E / 0W) - Show history - Download (1.074 MB)

- Mail - [02:01:18 - 02:01:18] - [Successful] (0F / 0E / 0W) - Show history - Download (743 B)

[+] build dvd-index (2) - [01:54:39 - 01:55:14] - [Successful] (0F / 0E / 0W) - Show history - Download (1.692 KB)

Figure 8-8: Project selection for deployments

Next to the project name is information on the number of log files present for the project and when they were created. Clicking on the plus sign [+] before each project name displays the deployments in descending order. Clicking on the [Show history](#) link next to each entry displays the associated log file. Clicking on the [Download](#) link allows the user to download the associated log file.

The [Change](#) link next to the project name is used to select a different project or deployment for viewing.

Detailed documentation on the history log view layout can be found in Chapter 8.3.1.2, starting on page 460.



8.4 Scheduling

8.4.1 Scheduling – Overview

This page provides an overview of all schedules.

Overview

Schedules:

<u>Name</u>	<u>Project name</u>	<u>Type</u>	<u>Status</u>	<u>last execution</u>	<u>Last duration</u>	<u>next update</u>
Repair references	Dokumentation WebClient (PRODUKTIV)	fixed	active	22.03.2012 10:12	1s 954ms	unknown
Generate WebEdit Tooltips	Dokumentation WebClient (PRODUKTIV)	daily	active	16.07.2012 18:00	38s 875ms	17.07.2012 18:00
Generate PDF Document	Dokumentation WebClient (PRODUKTIV)	daily	active	16.07.2012 18:00	1m 15s 865ms	17.07.2012 18:00
generate (daily)	Dokumentation WebClient (PRODUKTIV)	daily	active	16.07.2012 18:00	1m 10s 85ms	17.07.2012 18:00

Figure 8-9: Server schedule overview

All schedules configured on the server are listed in a table (see Chapter 7.5.1 page 372). In addition to the project name, the interval at which the schedules are carried out and their current status are displayed. Information on the period of the last scheduled execution and its duration as well as the period of the next scheduled execution are also displayed.

This list can be sorted by any existing column.



Project administrators see only the schedules of projects for which they are registered as administrator.

Clicking on a listed project name displays a more detailed overview of that project (see Chapter 8.2.1.1, starting on page 454). Clicking on the schedule entry name displays detailed information on that schedule entry.

Below the first table is another table listing schedule entries, e.g. deployments that have taken place within the last 48 hours.



Executed schedules in the past 48 hours:

Name	Project name	Status	Start	End	Duration
generate (daily)	Online-Dokumentation FirstSpirit...	active	16.07.2012 18:00:00	16.07.2012 18:04:27	4m 27s 280ms
build dvd-index	Online-Dokumentation FirstSpirit...	active	16.07.2012 18:00:00	16.07.2012 18:00:29	29s 100ms
generate (daily)	Dokumentation WebClient (PRODUKTIV)	active	16.07.2012 18:00:00	16.07.2012 18:01:10	1m 10s 85ms
Generate PDF Document	Dokumentation WebClient (PRODUKTIV)	active	16.07.2012 18:00:00	16.07.2012 18:01:15	1m 15s 865ms
Generate WebEdit Tooltips	Dokumentation WebClient (PRODUKTIV)	active	16.07.2012 18:00:00	16.07.2012 18:00:38	38s 875ms
generate full	Online-Dokumentation FirstSpirit...	active	16.07.2012 01:54:39	16.07.2012 02:01:18	6m 39s 106ms
generate full	Dokumentation WebClient (PRODUKTIV)	active	16.07.2012 01:54:39	16.07.2012 01:55:45	1m 6s 410ms

Figure 8-10: Scheduling – Executed schedules in the past 48 hours

In addition to the name of the schedule entry, the table provides information on the status of the executed actions as well as the name of the project for which the schedule entry ran. The schedule entry execution start and end times as well as the duration of the last execution are also displayed.

Log files for the individual deployment schedule entries can be accessed from the menu item "Log files / by deployment" (see Chapter 8.3.3 page 464).

8.4.2 Scheduling – Planned schedules

This page provides an overview of the planned schedules. A maximum of 25 schedules can be displayed at once. The arrow buttons above the table can be used to navigate between the individual pages.

All schedules that are still to be executed are listed in the table. In addition to the project name, the table shows information on the schedule ID and the type of action scheduled. The table also shows at what interval the schedules are to be carried out and when the next execution is scheduled.

This list can be sorted by any existing column.

Clicking on the listed project name displays detailed information on that project. (See Chapter 8.2.1.1, starting on page 454.)

8.4.3 Scheduling – Executed schedules

This page provides an overview of the schedules already executed. A maximum of 25 schedules can be displayed at once. The arrow buttons above the table can be used to navigate between the individual pages.

All schedules that have already been carried out are displayed in a table.




In addition to the name of the schedule entry, the table provides information on the name of the project for which the schedule entry ran. The table also shows the schedule entry execution start time, status, errors, users and duration of the last execution. This list can be sorted by project name, schedule entry name or the start of execution.

Clicking on the listed project name displays detailed information on that project. (See Chapter 8.2.1.1, starting on page 454.)

Clicking on the "Show history" link allows the user to view the log output for the schedule entry execution period. The output of the log details can be configured using checkboxes.

For instance, it is possible to set different log levels using **Log level**. If, for instance, the user wants to find error messages while generation is running, the user can set the log level to "Error" (for more information, see Chapter 4.3.6 page 103).


The maximum number of **Lines** to be displayed in the log output can also be configured. The view is refreshed by clicking on the  button.



Clicking on this button interrupts the line output.



Clicking on this button allows the user to expand the line-restricted output.

In addition to filtered output, it is possible to search directly for particular terms within the log. Once the user clicks on the  button, the search results are displayed at the bottom of the page under **Search results**.



8.5 Users

8.5.1 Users – Find

This page shows a sorted list of all users registered on the server. A maximum of 25 users can be displayed at once. The arrow buttons above the table can be used to navigate between the individual pages.

In addition to the user name, the table also provides information on the abbreviation, last login and whether the user is an LDAP user. The e-mail address and phone number are also included.

The search function can be used to limit the selection of displayed users. Words or partial words by which the user name column is searched can be entered in the search field. Clicking on the

 button initiates the search.

Clicking on a user displays detailed information about that user. This also includes information that is specified when the user is first added (see Chapter 7.2.4.1, starting on page 235) as well as a list of all projects on the server to which the user has access. The user's group membership to a project is specified for each project. Clicking on a different project for the selected user displays a detailed overview of this project. (See Chapter 8.2.1.1, starting on page 454.)

Clicking on the user's e-mail address opens an e-mail window where a message can be prepared and sent directly to the user.



8.6 FirstSpirit

8.6.1 FirstSpirit – Configuration

The configuration of the FirstSpirit server, the database connection, login procedure and some other settings are made in special configuration files in the FirstSpirit server installation directory. Editing these configuration files directly is not recommended (see Chapter 4.2 page 30).

FirstSpirit ServerManager features a convenient option for editing the configuration settings to configure the database connection (fs-database.conf) and login procedure (fs-jaas.conf) (see Chapter 6 page 194). Additional settings can be configured using the JMX console (see Chapter 9 page 494).

FirstSpirit ServerMonitoring features the following configuration options:

- | | | |
|-----------------------|---------------------------|--------------------------|
| • Server | (file: fs-server.conf) | Chapter 8.6.1.1 page 471 |
| • License | (file: fs-license.conf) | Chapter 8.6.1.2 page 471 |
| • Logging | (file: fs-logging.conf) | Chapter 8.6.1.3 page 472 |
| • System | (no configuration) | Chapter 8.6.1.4 page 473 |
| • Start options | (file: fs-wrapper.conf) | Chapter 8.6.1.5 page 473 |
| • Web applications | (file: fs-webapps.xml) | Chapter 8.6.1.6 page 474 |
| • Services | (config. system services) | Chapter 8.6.1.7 page 475 |
| • Login configuration | (file: fs-jaas.conf) | Chapter 8.6.1.8 page 476 |




This menu level is only available to server administrators.





8.6.1.1 Server

The FirstSpirit server is configured in the configuration file `fs-server.conf`, which is located in the FirstSpirit server installation directory (see Chapter 4.3 page 32).

The file can be opened for editing by clicking on the "Server" entry. The individual parameters for configuring the FirstSpirit server are explained in Chapter 4.3.1 ff. Some changes may require restarting of the server (e.g. if changing a port); other changes can be made while the server is running. The changes must be saved and the configuration file needs to be reloaded on the server.


Clicking on the  button saves the current server configuration changes and integrates them into the running server operation.


Clicking on the  button takes the user back to the configuration overview. Any recent changes to the configuration file are reset if the configuration file was not saved beforehand.

Restart server: if this option is selected, clicking on the  button will save the latest server configuration changes and then restart the server. (The FirstSpirit server starts up with help from the `fs-wrapper.conf` configuration file; see Chapter 8.6.1.5).

8.6.1.2 License

The `conf` subdirectory of the FirstSpirit server contains the `fs-license.conf` file. The file contains the current FirstSpirit license and can be viewed as needed by clicking on the "License" entry in FirstSpirit ServerMonitoring. The associated parameters are explained in Chapter 4.3.5, starting on page 101.

If a new license is to be installed on the server, the entire content must be added to this site and without modification. Clicking on the  button saves the new license file.

Clicking on the  button takes the user back to the configuration overview. Any recent changes to the license file are reset if the license file was not saved beforehand.

Changes to the configuration file are updated automatically on the server during operation within the specified time intervals. The FirstSpirit server therefore does not have to be restarted.





Tampering with the `fs-license.conf` file will invalidate the license. If changes become necessary (e.g. changing the IP address), please contact the manufacturer.

8.6.1.3 Logging

The FirstSpirit server `conf` subdirectory contains the `fs-logging.conf` file, which contains important configuration settings for the "Log" schedules and must be adjusted when necessary. The associated parameters are explained in Chapter 4.3.6, starting on page 103.

Any errors or notifications are passed to the logging system "log4j"³⁶ by default. The framework is used to carry out quantification of the log outputs. The available categories (Log level) are `DEBUG`, `INFO`, `ERROR`. Additional categories can also be configured if required (e.g. `FATAL`, `WARN`). The exceptions are the two levels `ALL` and `OFF`, which either completely deactivate logging (`OFF`) or output all messages without filtering them (`ALL`). Filtering and the type of output can be configured during runtime.

In addition to this default log, other log files can also be configured. The status of each log file (active | inactive) appears next to the name of the corresponding log file. An inactive log file can be activated by clicking on the **Use** button. The log file that was active previously is then deactivated, since only one log file can be active at any given time.

Clicking on the **Edit** button allows the user to make changes to the log file.

Clicking on the **Save** button saves the changes. Changes to the configuration file are updated automatically on the server during operation within the specified time intervals. The FirstSpirit server therefore does not have to be restarted.

Clicking on the **Cancel** button takes the user back to the configuration overview. The latest changes to the configuration file are reset if the file was not saved beforehand.

³⁶ More information is available at <http://logging.apache.org/log4j/>



8.6.1.4 System

All relevant server system information can be found on this page.



The screenshot shows a web interface titled "System". It has two tabs: "System information" (selected) and "File download". Under "System information", there are two sections: "Operating system" and "System properties".

Operating system	
Java virtual machine	Java HotSpot(TM) Server VM 20.2-b06
Java Runtime	Java(TM) SE Runtime Environment 1.6.0_27-b07
Operating system	Windows 7 () x86 6.1
Processors	4
Current time	Dienstag, 17.07.2012 17:06:01
Working directory	D:\Firstspirit_Server\Firstspirit_5

System properties	
backup_files	50
cmsroot	D:\Firstspirit_Server\Firstspirit_5

com.sun.management.jmxremote

Figure 8-11: ServerMonitoring – System information and file download

Operating system: information on the basic operating system is shown here, e.g. the JDK in use, the server operating system and number of processors.

System properties: the system properties (information on the directory paths, port numbers, class path, etc) are arranged here in a list.

The system properties shown are not configured in this overview; they are for informational purposes only.

System information: additional system information can be viewed by clicking on the "System information" button.


File download: Clicking on the "File download" button opens the file selection dialog of the server log directory (.../server/log). The server log files (fs-server.log) and other log files that are not available via ServerMonitoring can be downloaded here (fs-gc.log, for example).


8.6.1.5 Startup options


The FirstSpirit server `conf` subdirectory contains the file `fs-wrapper.conf`, which contains important configuration settings for the server startup and must be adjusted as necessary. The associated parameters are explained in Chapter 4.3.2, starting on page 74.



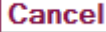
The configuration file is responsible for starting and stopping the Java process. Depending on the configuration, starting VM can be defined as necessary and the corresponding log outputs can also be written to the default output.


Clicking on  allows the user to edit the existing configuration. A text editor window opens where the content of the `fs-wrapper.conf` configuration file can be edited.

 Clicking on the button again closes editing mode and returns the user to the list overview.

Clicking on  saves the changes to the `fs-wrapper.conf` configuration file. If invalid configuration settings have been made, configuration errors will be indicated in ServerMonitoring during the saving process and saving will be canceled:

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

Clicking on the  button takes the user back to the configuration overview. The latest changes to the configuration file are reset if the file was not saved beforehand.


Restart server: if this option is selected, clicking on the  button will save the latest server configuration changes and then restart the server.




The server must be restarted any time changes are made to the `fs-wrapper.conf` configuration file.

8.6.1.6 Web applications

The FirstSpirit server `conf` subdirectory contains the `fs-webapp.xml` file, which contains important configuration settings for the internal web server and must be adjusted as necessary. The associated parameters are explained in Chapter 4.3.7, starting on page 106.

Clicking on the  button saves the changes to the `fs-webapp.xml` file.

Clicking on the  button takes the user back to the configuration overview. The latest changes to the configuration file are reset if the file was not saved beforehand.





To ensure that the changes are updated on the server, Jetty needs to be restarted. After saving, a message appears with the link in the menu item "Control / Web Applications" (see Chapter 8.6.2.2 page 479). Jetty can be restarted from there (within the ServerMonitoring environment).

8.6.1.7 Services

FirstSpirit system services can be configured using this entry. A service is a server component that can be activated via a public interface composed of input components or scripts. (Examples include spell checking or the CMS_INPUT_PERMISSION permissions input component service; see Chapter 7.3.11 page 265.)

The table lists all of the services available on the server.

Overview of configurable services:

Name: name of the system service. Clicking on the entry opens another window with configuration options for the particular service (see "Configuring a service" below).


Comment: optional comment for the service.


Type: name of module with which the service is associated. System-designated services are part of the FirstSpirit standard system module.

Autostart: automatic starting of a service can be activated or deactivated here. The function can be used like the "Activate autostart" or "Deactivate autostart" option in ServerManager (see Chapter 7.3.11 page 265).

Configuring a service:

File name: service file name. Clicking on the file name opens another window where a file, such as the PermissionService groups.xml file, can be edited.

Clicking on the  button saves the changes in the file.

Clicking on the  button takes the user back to the file overview. Any recent changes are reset if the file was not saved beforehand.

Size: file size.




Last change: date of last file version saved.


Create directory: this button is used to create a directory for the particular service. The directory is created in the file system under the module directory with the given name.

Create file: this button is used to create a file for the particular service. The file is created in the file system under the module directory with the given name.

8.6.1.8 Login configuration

The FirstSpirit server `conf` subdirectory contains the `fs-jaas.conf` file, which contains important configuration settings for the login procedure and must be adjusted as necessary. The associated parameters are explained in Chapter 4.3.4, starting on page 85.

Clicking on the  button saves the changes to the `fs-jaas.conf` file. The file is updated on the FirstSpirit server automatically. The server does not have to be restarted.

Clicking on the  button takes the user back to the configuration overview. Any recent changes to the configuration file are reset if the file was not saved beforehand.



8.6.2 FirstSpirit – Control

The Control menu item is used for control after the server configuration is changed or the FirstSpirit server is updated.



This menu level is only available to server administrators.

8.6.2.1 Maintenance mode

This function is used to activate or deactivate FirstSpirit Server maintenance mode. Maintenance mode can be implemented:

- for scheduled updating of the FirstSpirit server (in order to shut down the server properly while it is running).
- in order to limit access to projects (e.g. in the case of conversions or larger updates).

Maintenance mode is configured in the ServerManager server properties. To do this, the "Maintenance mode" action must be assigned to a schedule entry (via schedule management) (for information on configuration, see Chapter 7.5.9.5 page 397).

Using FirstSpirit ServerMonitoring, it is possible to:

- start existing maintenance mode schedule entries,
- view the current status of maintenance mode schedule entries, and
- stop running maintenance mode schedule

entries.



The screenshot displays the FirstSpirit Administration interface. The top navigation bar includes the FirstSpirit logo, a language dropdown, and user information (User: Admin, End monitoring). A status bar shows system details: FirstSpirit 5.0.100.52761, Projects 4, Actions 0, Memory [], Sessions 2, Preview 0, Load []. The main navigation menu on the left includes Overview, State, Activities, Sessions, Projects, Logfiles, and Scheduling. The 'Maintenance mode' section is active, showing a dropdown menu with 'Wartungsauftrag' selected and a 'start' button. Below this is a table titled 'Active maintenance periods' with the following data:

Schedule Entry	State	Projects	Next step	Maintenance period	Actions
Wartungsauftrag	In preparation Inactive	all	Show first warning Aug 23, 2012 1:40:22 PM	Aug 23, 2012 1:50:22 PM Duration: 1 minutes	End maintenance mode

A 'Refresh' button is located below the table.

Figure 8-12: ServerMonitoring – Control – Maintenance mode

start: if a maintenance mode schedule entry has been created (see Chapter 7.5.9.5 page 397), it can be selected from the "Maintenance mode schedule entry" drop-down list and started using the "start" button. The message "Maintenance mode started" appears. This button can also be used to start a schedule entry multiple times simultaneously.

Scheduled maintenance periods: here the scheduled maintenance mode schedule entries are displayed with the settings that were made in the server properties.

Active maintenance periods: here the running maintenance mode schedule entries are displayed with the settings that were made in the server properties.

Status: this column shows the current status of the schedule entry (to view the updated status, click the "Refresh" button). Depending on the settings in the schedule entry, the statuses are displayed in the following order:

- *In preparation | Inactive:* the schedule entry is started, but still does not affect the user.
- *In preparation | Show first warning:* in this status, the message "Maintenance work will be carried out on FirstSpirit Server | on this project in x minutes. Estimated duration: approx. y minutes." appears in the clients and, if applicable, on the start page as a warning. See the "Display notification after" option in the schedule entry (Chapter 7.5.9.5 page 397).
- *In preparation | Session end message:* in this status, the message "FirstSpirit Server | Project is unavailable due to maintenance work starting 21.08.2012 15:15:00. Estimated duration: approx. y minutes. Please terminate your session." appears in the clients and, if applicable, on the start page. See the "Show End Sessions warning after" option in the schedule entry (Chapter 7.5.9.5 page 397).



- *In preparation | Reject sessions*: in this status, no more logins are possible. See the "Refuse new sessions after" option in the schedule entry (Chapter 7.5.9.5 page 397).
- *Active | Maintenance mode*: maintenance mode is active in this status. Maintenance work can now be performed and, if necessary, the FirstSpirit server can be shut down. See the "Start maintenance mode after" option in the schedule entry (Chapter 7.5.9.5 page 397).
- *End*: logging onto the server is possible again in this status. See the "Estimated duration" option in the schedule entry (Chapter 7.5.9.5 page 397).

Projects: this column shows which projects are affected by the maintenance mode schedule entry (the "Apply to projects" option in the schedule entry; see Chapter 7.5.9.5 page 397).

Next step: this column shows what the next step is and when it will begin. The steps are described under "Status" (see above).

Maintenance period: this column shows when the maintenance period begins (based on the schedule entry start time and the times for steps 1 through 4) and the duration (the "Estimated duration" option in the schedule entry; see Chapter 7.5.9.5 page 397). Depending on the configuration of the options "Refuse start tasks" and "End sessions during maintenance, refuse new sessions", schedule entries that occur within the maintenance period are halted and no server or selected project logins are possible during the maintenance period.

End maintenance mode: clicking on this link terminates the relevant schedule entry. The message "Maintenance ended" appears in ServerMonitoring. The message "Maintenance work on FirstSpirit Server has been completed" is output on the clients and the start page.

Refresh: this button refreshes the view.

8.6.2.2 Web applications

The web applications configured on the server are displayed in this view (see Chapter 7.3.13 page 278). The FirstSpirit web applications (i.e. *fs5root*, *fs5preview*, *fs5staging*, *fs5webedit*, *fs5webmon*) are included by default.

Restart: click on this link to restart the relevant web application.

Restart Jetty: click on this button to restart the internal Jetty. Restarting is required, for instance, so that changes to the `fs-webapp.xml` file can be updated on the server (see 8.6.1.6 page 474).



8.6.2.3 Update



This menu level is only available to server administrators.

In addition, the update functionality is only available if the FirstSpirit server was started using the Java wrapper (for configuration, see Chapter 4.3.2).

The function is used to update the version of the FirstSpirit Server software.

The screenshot shows the 'Update' tab selected in the top navigation bar. Below the navigation bar, the title 'Update' is displayed in a large, bold font. Underneath, there is a section labeled 'Current server version:' followed by the value '5.1.12.345'. Below this, there is a section for file selection. It includes a label 'fs-server.jar:' followed by a text input field and a 'Durchsuchen...' button. Below the input field, there is an 'Add module file' button and a 'Start update' button.

Figure 8-13: ServerMonitoring – Control – Update (initial view)

Browse: clicking on this button opens a window where a new jar file (fs-server.jar) or a new module file (.fsm) can be selected. The files are selected through the local file system.

Add module file: each time the user clicks on this button, a new input area is added for a module file. The module file is selected the same way the fs-server.jar file is selected using the "Browse" button.

Note: this area is not used to install modules. This means that only modules already installed on the server can be updated. Module files added here for which no earlier version was installed on the server are not installed or updated.

Delete: files already selected can be removed by clicking on this button.



Figure 8-14: ServerMonitoring – Control – Update (after file selection)

Start update: after the files have been selected, the server can be updated by clicking the "Start update" button. After clicking on the button, the files are first uploaded and then the system undergoes an integrity and signature check. If this is successful, the server is shut down and then restarted automatically.



Any client connections to the server should be terminated before the server update to prevent any loss of data. Maintenance mode can be used to do this (see Chapter 8.6.2.1 page 477).

Before the server is shut down, the files are temporarily saved under:

~\server\update\server\lib\ or

~\server\update\data\modules\update\

and then loaded from there when the server restarts. When the update is successful, the files are automatically removed from these directories. If the update fails, the jar and fsm files remaining in the directories can be deleted manually (note: do not remove the fs-update.conf and fs-update.jar files).

Updating installed web applications: all installed FirstSpirit web applications under the ~fs5\web directory are under the control of server version management and will be rolled out again when the server is started. If the particular application directory (under ~fs5\web) contains a `version.txt` file, the system will first check if a newer version of the installation is present before updating by using the version number in the `version.txt` file. If there is a new version,



the new version will be rolled out. The web server control then handles the update (and restart) of the web application on one or more web servers (except for external web servers; see Chapter 7.3.12 page 271).

Updating in a cluster group: if the updated FirstSpirit server is a cluster master server in a cluster group, the participating cluster slave servers will also be updated and restarted. This means that if the master server is shut down, all slave servers will also be shut down and restarted after a brief 60 second wait.

Note: updating the slave servers is only carried out for the default configuration if the cluster master server and cluster slave servers use a shared file system (for more on clustering, see Chapter 7.3.14 page 284). In this case, the slave servers access the master server's `fs-server.jar` file and are thus also updated when the master server is updated. To prevent conflicts during the update process, the slave servers run in a save state using their own `update.jar` file while the master server is restarted. This means that the shared `fs-server.jar` is not being used when the master server is restarted and can therefore be updated.



When updating to a different major version of the software, the `fs-license.conf` license file also needs to be replaced. In addition, it is important to read the version update release notes beforehand.

Using the ServerMonitoring control network function, additional servers can be selected for updating (see Chapter 8.6.2.7 page 484).

8.6.2.4 Services

This function is used to control the FirstSpirit server services. A service is a server component that can be activated via a public interface composed of input components or scripts. Examples include spell checking or the permissions input component service. The system module with the default services (e.g. the PermissionService) is already included as standard in FirstSpirit and is available after the software is installed. However, additional services subsequently installed on the server can be displayed as well (see Chapter 7.3.11 page 265). Services always apply system wide.

Name: name of the service.

Comment: description of the particular service.



Type: name of module with which the service is associated.

Start/Stop: function for starting/stopping the service. If the service has already been started, the only option available is "Stop". If the service has not been started, the only option available is "Start". This function is similar to starting and stopping a service through FirstSpirit ServerManager (see Chapter 7.3.11 page 265).

Reboot: function for restarting the service. Unlike simple start or stop functions, the service that is already started is first stopped and then automatically restarted.



After updating modules that have dependencies to modules with services, these services need to be restarted manually.

8.6.2.5 Restart server

Restart server: clicking on this button restarts the FirstSpirit server.



No confirmation prompt appears before the server is restarted. Any client connections to the server should be terminated before the restart to prevent any loss of data. Maintenance mode can be used to do this (see Chapter 8.6.2.1 page 477).

8.6.2.6 AppCenter licenses

Use of the FirstSpirit AppCenter is subject to a new licensing model. Unlike the previous FirstSpirit (module) add-on licensing, it is not the functionality, but the number of integrated applications that is licensed. The license.APPTAB_SLOTS license parameter defines how many different application integrations can be used (see Chapter 4.3.5 page 101).

This view shows the type and number of applications that are currently registered using the license.APPTAB_SLOTS parameter.

The **Reset usages** button is used to reset the number of registered applications to 0, if necessary. Registered applications that are currently open in clients can continue to be used until the application or the associated tab is closed. The server does not need to be restarted.



8.6.2.7 Network

On this page, all FirstSpirit servers within the current network are listed. The FirstSpirit version and additional information are provided for each server and are obtained from the values in the server configuration file `fs-server.conf` (FirstSpirit Server host name, socket port, HTTP port) and license settings (license type).

The screenshot shows the 'Network' tab in the FirstSpirit administration interface. It displays a table of servers with columns for Server, HTTP port, Version, Type, and Socket port. The table lists several servers, including '4.2 Stable Release', '4.2R4 Stable Release', '5.0 SVN-Version', 'FirstSpirit 4.2R4 (Produktion)', and 'e-Spirit Intranet (Produktiv)'. Below the table, there is a list of modules installed on the selected server, such as 'Apache FOP', 'Apache FOP v0_20_5', 'ContentDependentMediaRelease', 'Exalead-Push', 'ExcelAddressImporter', 'FIRSTpersonalisation', 'FirstSpirit FormEdit', 'FirstSpirit Geolocation Development', 'FirstSpirit Office', 'Intranet - FSDev', 'JSTL', 'Perf4J', 'PostgreSQL_JDBC_Driver_8_4', 'SpellService', and 'lastChanges'. At the bottom of the table, there is a button labeled 'Update FirstSpirit server'.

Server	HTTP port	Version	Type	Socket port
[+] 4.2 Stable Release		4.2.454.47473	production	
[+] 4.2R4 Stable Release		4.2.454.47473	production	
[+] 5.0 SVN-Version		5.0_BETA.33.51813	production	
[+] FirstSpirit 4.2R4 (Produktion)		4.2.472.51994	production	
[+] e-Spirit Intranet (Produktiv)		5.0_BETA.34.52001	production	
Apache FOP		5.0_DEV.19_49938		
Apache FOP v0_20_5		5.0_DEV.19_50021		
ContentDependentMediaRelease		0.0.1_129		
Exalead-Push		2.0		
ExcelAddressImporter		1.0.20		
FIRSTpersonalisation		5.0_DEV.19_50179		
FirstSpirit FormEdit		5.0_DEV.19_49938		
FirstSpirit Geolocation Development		5.0_DEV.19_47664		
FirstSpirit Office		5.0_DEV.19_49938		
Intranet - FSDev		1.3_5.0_DEV.25_2564		
JSTL		1.1		
Perf4J		0.9.13		
PostgreSQL_JDBC_Driver_8_4		8.4.701		
SpellService		5.0_DEV.19_49938		
lastChanges		1.0-SNAPSHOT		
[+] fs42build (Intranet/Internet)		4.2.453.46978	production	

Update FirstSpirit server

Update network

Clicking on [+] expands the entry. Information on the modules installed on the particular FirstSpirit server is then provided. User developed modules are **not** displayed.

Update network The displayed server information can be updated using this button.

This and other information can also be viewed using JMX monitoring and is described in more detail in Chapter 9.6 page 503.

Update FirstSpirit server The "Update FirstSpirit server" button appears in each expanded server entry. This function can be used to update the version of the FirstSpirit Server software (via FirstSpirit ServerMonitoring). Clicking on the button opens the ServerMonitoring web application for the selected FirstSpirit server. After successful authentication by the server



administrator, the FirstSpirit Control – Update area is displayed for selecting the fs-server.jar file and the desired module files (see Chapter 8.6.2.3 page 480).

8.6.3 FirstSpirit – Message

A text message can be edited on this page and sent to all active clients (start page, ServerManager, ServerMonitoring, SiteArchitect and ContentCreator) on the server in the form of a pop-up window.



This menu level is only available to server administrators.

8.6.4 FirstSpirit – Databases

FirstSpirit stores the highly structured content of the data store in a database. All information on connecting databases to the FirstSpirit server is available in Chapter 4.9, starting on page 155.

8.6.4.1 Overview

Overview of all currently configured database connections on the FirstSpirit server.

Configuration of a database connection is handled in the `fs-database.conf` configuration file. A description of the parameters can be found in Chapter 4.3.3, on page 84. Changes to the `fs-database.conf` configuration file can be made through the FirstSpirit ServerManager (see Chapter 7.3.5 page 252).



Only the server administrators can see the `PASSWORD` and `USER` parameters.

8.6.4.2 Status

The "Status" menu item shows more information about connected databases. Below the displayed database connection are the projects that access this database.

The link to the project below the database layer is used to call up the project detail page (see Chapter 8.2.1.1 page 454).



8.6.5 FirstSpirit – Monitoring

8.6.5.1 Memory

This page shows a graph of the memory consumption over the past 24 hours. The black line indicates the defined `-Xmx` value from the configuration file `fs-wrapper.conf` (see Chapter 4.3.2.3, parameter `wrapper.java.maxmemory`). The light purple section is the area reserved in the operating system by Java VM for the Java heap. The dark purple section is the actual heap used.

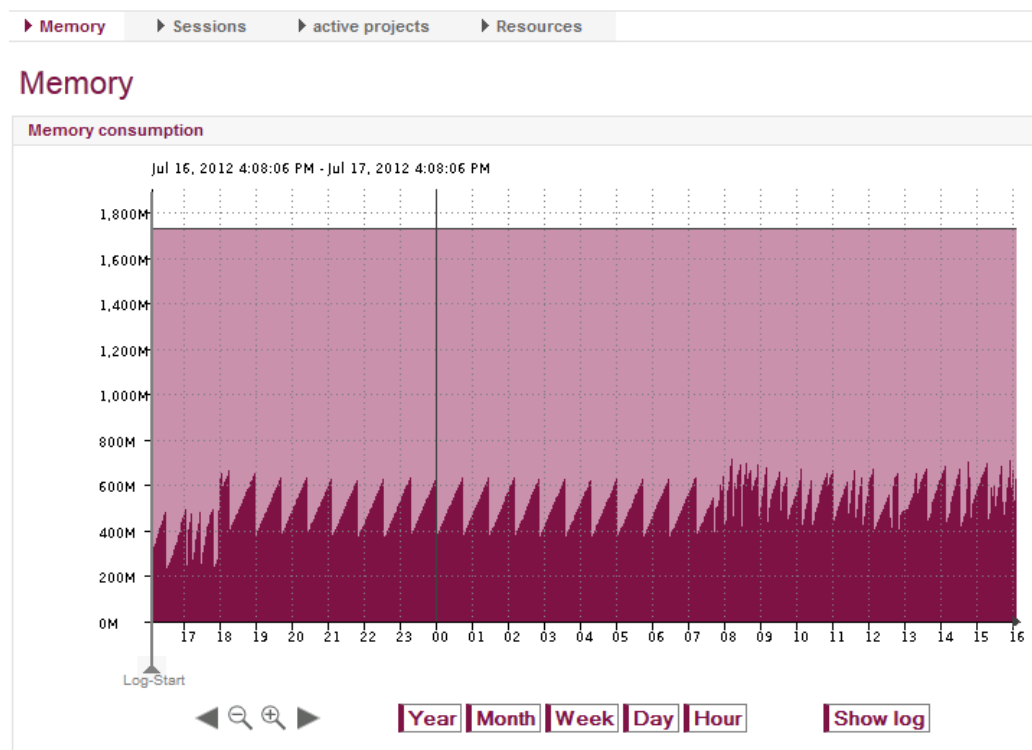






Figure 8-15: Monitoring – Memory consumption





The period covering the memory consumption shown can be changed as desired using the buttons **Year**, **Month**, **Week**, **Day** and **Hour**. Likewise, the  button shifts the period by one category towards "Year", and the  button shifts the period by one category towards "Hour". The arrow buttons can be used to move the displayed period by one year (month, week, etc.) in the past  or in the future , depending on the category set.


Clicking on the button **Show log** displays the server log file for the selected period. (See Chapter 8.3.1.2, starting on page 463.)



8.6.5.2 Sessions





This page shows a graph of the number of sessions that were simultaneously active on the server within the last 24 hours.


The period covering the active sessions shown can be changed as desired using the buttons **Year**, **Month**, **Week**, **Day** and **Hour**. Likewise, the  button shifts the period by one category towards "Year", and the  button shifts the period by one category towards "Hour". The arrow buttons can be used to move the displayed period by one year (month, week, etc.) in the past  or in the future , depending on the category set.

Clicking on the  button displays the server log file for the selected period. (See Chapter 8.3.1.2, starting on page 463.)

8.6.5.3 Active projects

This page shows a graph of the number of projects that were simultaneously accessed within the last 24 hours.

The period covering the active sessions shown can be changed as desired using the buttons **Year**, **Month**, **Week**, **Day** and **Hour**. Likewise, the  button shifts the period by one category towards "Year", and the  button shifts the period by one category towards "Hour". The arrow buttons can be used to move the displayed period by one year (month, week, etc.) in the past  or in the future , depending on the category set.

Clicking on the  button displays the server log file for the selected period. (See Chapter 8.3.1.2, starting on page 463.)

8.6.5.4 Resources

This page covers the following information:

- Amount of memory (user memory, allocated memory)
- Hard disk space (used space, free space)
- Size of backup and export directories





This information is not determined automatically during startup of the FirstSpirit server; it is available only after the statistics are updated (until the next time the server is started). The value for the statistics update is configured in the global server properties (see "Daily statistics (time)" in Chapter 7.3.1 page 243) (the default setting is "0" hours).

The table lists all of the projects installed on the server. In addition to the project name, the table shows information on the project ID and the disk space. The table also shows the maximum amount of available disk space (quota) (if no limit is entered, the value -1 is shown). This list can be sorted by any existing column.

8.6.5.5 Threads



This menu level is only available to server administrators.

The page contains information on the monitoring of the system's current state.

Thread dump: clicking on this button generates and displays a current thread dump. Clicking on [+] expands the displayed level. The buttons at the top of the window can be used to analyze the current thread dump and display it in a prepared view.

Thread Name	ID	State	CPU Time	Block Count	Block Time	Count
"AppletIOServlet.sessionTimer"	id=47	TIMED_WAITING		1	-1	10
"BerkeleyDbBackend-0"	id=33	TIMED_WAITING	109200700	24	-1	8
"PoolThread[DEFAULT]-1"	id=24	WAITING	312002000	20	-1	4
"Attach Listener"	id=5	RUNNABLE		0	-1	3

Figure 8-16: Threads – View sorted by "Trace"

Trace: clicking on this button groups the current threads by 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. The prepared view makes it possible to determine relatively



easily which actions are currently in progress on the server.

Topline: clicking on this button groups the current threads by similarities within the first five lines. Similar threads are groups in the prepared view for an entry. The number on the right shows the number of similar threads found. The prepared view makes it possible to determine relatively easily at what stage of execution the threads are currently in. This view makes it easier to trace bottlenecks, such as when multiple threads are always wait at a particular location in the code.

Lock: clicking on this button analyzes the current threads by locks. If multiple threads are in the BLOCKED status (see "Status" below), they will wait for a different thread to release an object. If there are multiple objects constantly in a waiting status that are being processed at a similarly slow rate, this can be an indicator of a bottleneck.

Text: clicking on this button displays the entire unprocessed thread dump with all its information.

In addition to generating a thread dump, other dumps can be generated at particular time intervals. The total number of thread dumps desired can be entered in the first box, and the time intervals can be entered in the second input box. Clicking on the "Generate" button initiates generation.

Thread dumps at intervals of Minutes

Table: the table view of the page shows the status of the current threads. The thread information shown here represents a snapshot or detail of the current status:

ID: each thread has a unique thread ID for identification.

Name: name of the thread.

Status: the threads may have different statuses:

- NEW – newly generated thread that has not yet been started.
- RUNNABLE – the thread is being processed or is waiting to run (waiting for CPU).
- BLOCKED – the thread is waiting for a lock to be released.
- WAITING – the thread is waiting for another thread before it can continue.
- TIMED_WAITING – the thread is waiting for a specified time to elapse (e.g. after sleep() or wait() is called using Timeout).
- TERMINATED – the thread has been terminated.



IN (In Native): provides information on whether the thread is executing native code via the Java Native Interface (JNI) (true) or not (false).

SP (Suspended): provides information on whether a thread has been started (false) or not (true).

BC (Blocked Count): the number provides information on how often a thread was in the BLOCKED status during execution.

WC (Waited Count): the number provides information on how often a thread was in the WAITING status during execution.

Clicking on a thread within the table opens the thread's detail window. Here all table information is clearly listed and the lock object and owner are also displayed.

ID	3
Name	Finalizer
Status	WAITING
Lock	java.lang.ref.ReferenceQueue\$Lock@a7d346
Owner	-
In Native	false
Suspended	false
Blocked Count	240
Waited Count	32

Figure 8-17: Threads – Details

Lock: if the thread's status is BLOCKED, it is waiting for the lock object specified here to be released.

Owner: if a thread's status is BLOCKED, the name of the thread currently holding the lock on the object is output.

The other information is described within this Chapter (see above).



8.6.5.6 VM memory

This area shows information about the current Java VM memory utilization.



This menu level is only available to server administrators.

Name: description of the memory pool shown.

- **Code Cache:** memory pool used for the internal evaluation for processes such as compiling.
- **Eden Space:** memory pool where most objects are initially generated. As soon as the garbage collector (GC) clears the Eden space, the surviving objects are transferred to the survivor space.
- **Survivor Space:** memory pool where the transient objects from the Eden space were transferred after GC (garbage collection).
- **Tenured Gen:** memory pool for persistent objects that have existed for some time in the survivor space and have been transferred here.
- **Perm Gen:** memory pool for JVM objects that are permanently required.
- **Total:** no memory pool. Provides a general overview of all pools available.

Type: type of memory (HEAP || NON_HEAP).

Max: maximum amount of memory available (in bytes) for memory management.

Used: memory currently in use (in bytes).

Initial: initial, allocated memory (in bytes) when JVM is started.

Committed: guaranteed available memory (in bytes) for JVM.

In addition to the table overview in the top part of the window, a graphic representation is available for each memory pool.



8.6.5.7 Key data

Java Management Extensions (JMX) is a standard interface available for managing Java applications. It is possible to monitor and manage the FirstSpirit server using the JMX console (see Chapter 9 page 494).

Some information available as a graphic representation is also available within ServerMonitoring under the "Key data" entry.



This menu level is only available to server administrators.

TasksWaiting: number of actions already forwarded from the corresponding execution cue, but still waiting for execution in the internal thread pool (see Figure 9-6) (see Chapter 9.8.3 page 507).

TasksRunning: number of currently running actions (e.g. an indexing job or an event (see Chapter 9.8.3 page 507).

TasksQueued: if an action, such as indexing, is initiated internally, the corresponding task will first be placed in a queue (exception: high priority tasks) (see Figure 9-6). The value shown here provides the number of currently queued actions (see Chapter 9.8.3 page 507).

ExecutionRate: number of tasks executed within the last 60 seconds (see Chapter 9.8.3 page 507).

Average Preview Creation Count: number of pages or page references for which a new preview was calculated within the last 60 seconds (see Chapter 9.13 page 511).

Average Session Creation Rate: number of new sessions opened within the last 60 seconds (see Chapters 9.22.1 page 520).



8.6.6 FirstSpirit – Clustering

The "Clustering" menu is used to open an overview of the existing cluster nodes for generation. Initially the overview is blank (if only a FirstSpirit master server is present). Cluster nodes are displayed only after they are present in the cluster group (see Chapter 7.3.14 page 284).

Any number of "generation slaves" can be displayed for the distributed processing of generation schedule entries.

The following information is shown for each generation server:

Version: FirstSpirit Server version number.

Server type: the cluster node server type is specified here. The overview shows, for instance, the "SLAVE (Generation Slave)" type, which is responsible for deployment processes. Multiple generation servers can of course be used. For instance, multiple deployments can be distributed across different servers, if necessary.

Last contact: time of last server contact (ping).

Current status: the current status of the server:

- IDLE Idle
- BUSY The server is currently busy handling processes.

Load: shows the percentage of server utilization, such as for processing a generation schedule entry.



9 FirstSpirit JMX Console

Java Management Extensions (JMX) is a standard interface available for managing Java applications. It is possible to monitor and manage the FirstSpirit server using the JMX console. While FirstSpirit ServerMonitoring primarily concerns itself with manual monitoring, the JMX interface is used for automatic monitoring and is ideally integrated into any existing, company-wide monitoring system. The JMX console can basically also be used interactively.

Compared to ServerMonitoring (see Chapter 8 page 449), clearly more detailed information is provided about the JMX console. JMX offers a standardized way to view and manage application resources or resources under JVM. All values and operations are provided by what are known as managed beans (MBeans). Some MBeans and information provided by them are covered in the following Chapters (starting with Chapter 9.3).



The structure and function of the MBeans explained in the following are continually subject to change. Due to continuous updating, the documentation may only be suitable to a limited degree. It is therefore possible that due to the lag time between this publication and changes to JMX, figures or descriptions in this documentation may deviate from the current view of the JMX console interface.



9.1 Starting the JMX console

To use the JMX console effectively, a remote connection must exist to the application that will be monitored. Starting with JDK 1.5, it is possible for an integrated platform function to start a JMX agent. The `fs-server.conf` configuration file must be adapted in order to activate this function for FirstSpirit Server (for configuration information, see Chapter 4.3.1.18 page 67).

The JConsole is started in conjunction with the application using the "jconsole" command line call:



Figure 9-1: Starting the JConsole (remotely)

Remote access is started using the "Remote Process" option. In addition to the host name, the JMX port of the target application can also be entered here. If the corresponding parameters have been configured (recommended for production systems), authentication using the "User name" and "Password" is also required. The JConsole is started by clicking on "Connect". (The appearance of the JMX console interface depends on the JConsole used.)

An extensive description of the GUI is available at:

<http://docs.oracle.com/javase/6/docs/technotes/guides/management/jconsole.html>



9.2 MBeans

All values and operations for JMX monitoring of the FirstSpirit server are provided by managed beans (MBeans). These are under the "MBeans" tab of the JConsole (see Figure 9-2).

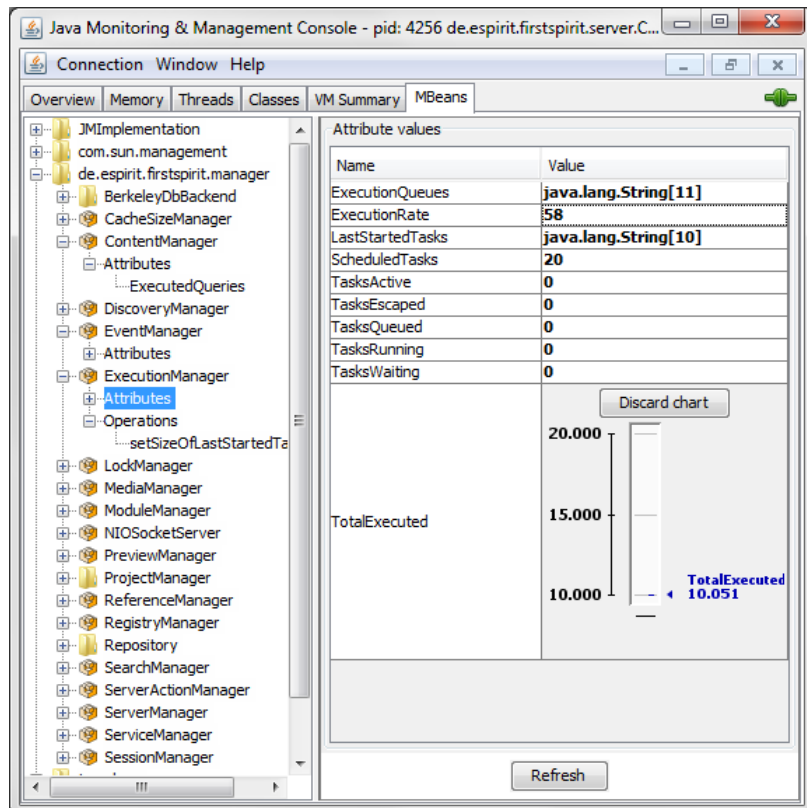


Figure 9-2: "MBeans" tab in JMX console

MBeans provide information and statistics ("attributes") as well as processing options ("operations").

Attributes: information and statistics that can be viewed in the form of a table (standard view) or chart (switch view by double-clicking on the corresponding value). Clicking on the "Discard chart" button closes the chart view.



Attribute values	
Name	Value
ActiveCalculated	0
ActiveDuration	432ms
ActiveQueued	1
ActiveRecalculations	3
ActiveRepaired	0
CalculationRate	1694

Figure 9-3: Information and statistics viewed in the JMX console (Attribute values)

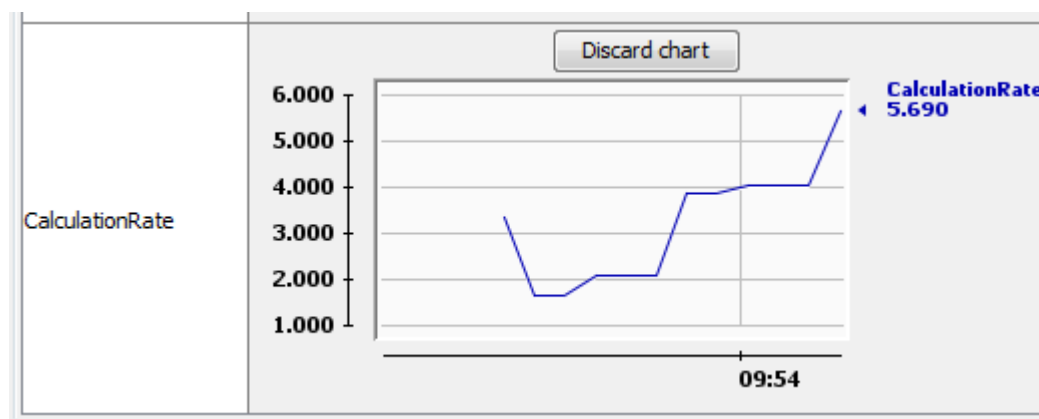


Figure 9-4: Statistics viewed in the JMX console (Attribute values - chart)

Operations: operations that can be called using specific managers in order to view additional statistics or initiate particular actions, for instance (see Figure 9-5). The operations are started by clicking on the relevant button.



Executing operations via the JMX console while the system is running may cause errors (e.g. when restarting the manager) and should only be done by experienced FirstSpirit administrators.

The operations are executed immediately (without prior prompting).

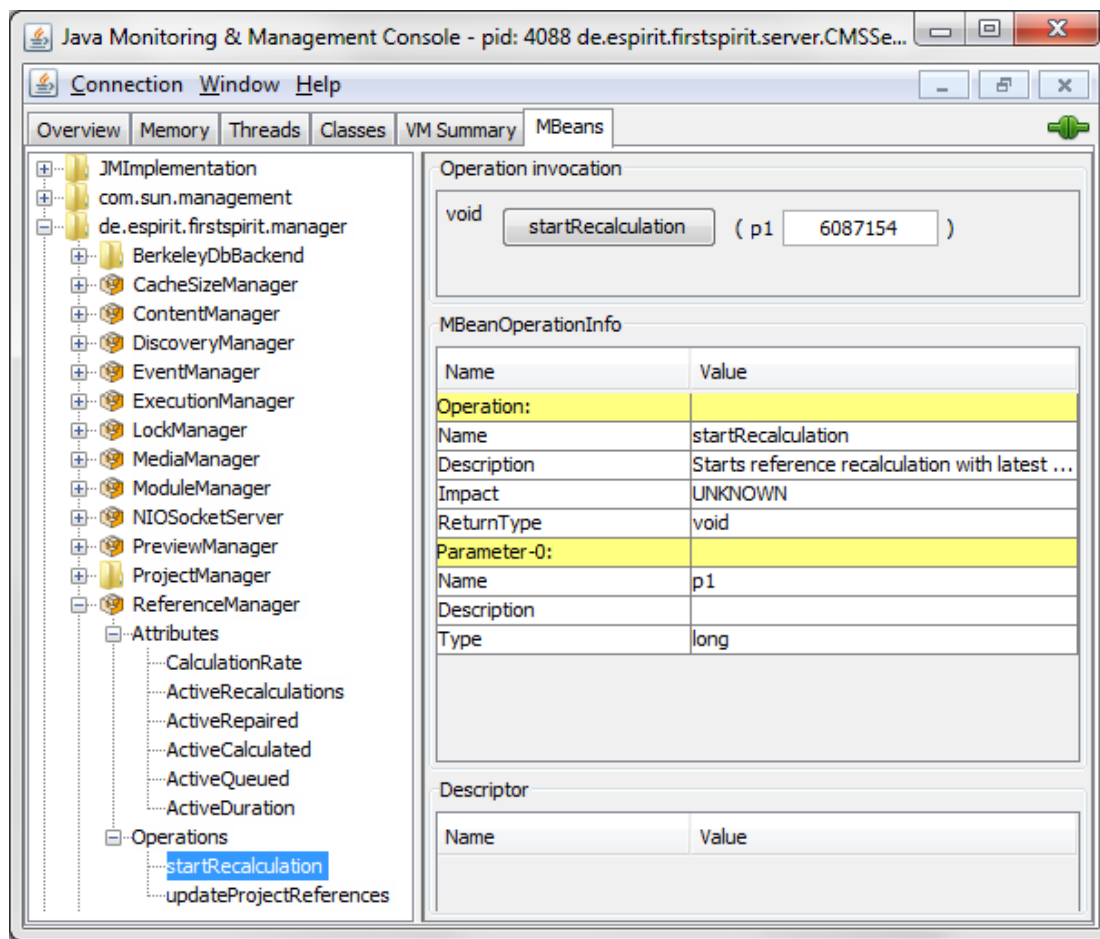


Figure 9-5: Reference recalculation via the JMX console (Operations)

Info: general information about the manager (name of manager and class implementing the manager).

The relevant information or operations are displayed only in managers in which they are relevant (for examples, see Chapter 9.3.1 page 499, and Chapter 9.3.2 page 500).

Some MBeans and information provided by them are covered in the following Chapters (starting with Chapter 9.3 ff.).



9.3 BerkeleyDbBackend

This entry shows information on the connected back end systems. Currently only the Berkeley DB back end is supported; therefore, only information about the Berkeley Storage Repository of the projects used is displayed in the JMX console. The view, as well as the Repository Manager (see Chapter 9.17) or Cache Size Manager (see Chapter 9.4), is dynamic. This means that the entry is loaded in the JMX console only after the corresponding project has been opened by at least one user. The information is then updated dynamically at runtime. Each project has its own MBean, which is displayed under the "BerkeleyDBBackend" folder.

9.3.1 Attributes

backupMode: Berkeley DB uses internal cleaning tools to remove obsolete data from the database at particular intervals. Backup mode is used to switch off the Berkeley DB cleaner during backup (value=true) (default value=false).

batchCleanerRunning: in addition to the Berkeley DB cleaning tool, an internal cleaner can be used to explicitly call the Berkeley DB cleaning tool.

cachePercent: specifies the cache for the back end as a percentage of the main memory.

cacheSize: specifies the cache size for the back end in bytes.

cleanedLogs: number of log files cleaned up to this point (Berkeley DB database files).

cleanerBacklog: number (queue) of log files not yet cleaned.

cleanerThreads: number of internal Berkeley DB threads responsible for cleaning. The number can change and depends on the value of `cleanerBacklogs`.

environmentHome: path to the directory where Berkeley DB data are stored (repository).

isReadOnly: information on whether the back end is write-protected (value=true) or not (value=false).



`isTransactional`: information on whether the back end is transactionally open (value=true) or not (value=false).

`lockTimeout`: timeout of Berkeley internal locks in μ s.

`openCursors`: number of database cursors open when using iteration via the database.

`runCleaner`: information on whether Berkeley DB cleaning is enabled (value=true) or not (value=false).

9.3.2 Operations

The **operations** displayed here can be executed on `cleanLog()`, for instance, by clicking on the corresponding button:

`java.lang.Integer cleanLog()`: executing this method removes obsolete log files (Berkeley DB database files) (return value: number of logs removed by this function).

`void evictMemory()`: executing this method frees up the Berkeley DB cache.

`void sync()`: executing this method writes all changed data to the hard disk that have not yet been saved.

In addition to operations, a number of statistics are available that can also be executed by clicking on the relevant field, such as on `getLockStats(...)`:

`void resetStats()`: executing this method resets all Berkeley DB statistical information.

`java.lang.String getEnvironmentStats(boolean)`: executing this method displays general statistics about the Berkeley DB.

`java.lang.String getLockStats(boolean)`: executing this method displays the lock statistics.

`java.lang.String getKeyDatabaseStats(boolean)`: FirstSpirit data are stored in different databases (key, value and blob databases). Executing this method displays the statistics of the key Berkeley DB.



`java.lang.String getValueDatabaseStats(boolean):` FirstSpirit data are stored in different databases (key, value and blob databases). Executing this method displays the statistics of the value Berkley DB.

`java.lang.String getBlobDatabaseStats(boolean):` FirstSpirit data are stored in different databases (key, value and blob databases). Executing this method displays the statistics of the blob Berkley DB.



It can take a moment for the call if the volume of data is particularly high.

9.4 Cache Size Manager

The Cache Size Manager splits the available main memory into different caches. This entry displays information on the overall cache size and information on the project-specific caches ("SharedCaches"). Like the Repository Manager (see Chapter 9.17), the Cache Size Manager is a dynamic manager. This means that the entry is loaded in the JMX console only after the corresponding project has been opened by at least one user. The information is then refreshed dynamically at runtime.

9.4.1 Attributes (all)

`CacheCount:` total number of currently active caches (for all projects).

`CacheSize:` absolute size of cache. The value shown here is a result of the values set for the parameter `CACHE_SIZE` (absolute) or `CACHE_PERCENT` (percentage) in the `fs-server.conf` configuration file (see Chapter 4.3.1.12 page 60).

`LastCalculationDate:` the last time the memory was redistributed among the caches.

`UsedCacheSize:` specifies the cache size currently used by all caches.

9.4.2 Operations

`void distributeCacheMemory():` executing this method redistributes the available cache memory. After execution, the value stored under "LastCalculationDate" is adjusted (see 9.4.1).



9.4.3 Attributes (project-based)

In addition to the overall CacheSize information, the projects have their own Cache Size MBeans, which are grouped under the MBean shared caches in the Cache Size Manager.

MissCount: the counter records the number of requests from all projects to the server that could not be handled directly from the cache. These requests must be forwarded to the back end and handled via access to the hard disk. A high or sharply increasing value is an indicator of poor cache usage and thus delayed response times.

PreferredSize: the preferred cache size is shown. This cache size is not always the same as the actual assigned value (size), depending on the "PreferredSize" distributed, among other things.

Size: by default, the Cache Size Manager allocates the size of a MB to each project cache. If the preferred size is higher, more memory may be provided. The amount of memory a project cache receives depends on the available global cache size, the preferred size of all projects and the weight for the individual projects.

Weight: the weight is a parameter defined in the project properties of ServerManager (see Chapter 7.4.2). A project cache receives more (if weight is higher) or less (if weight is lower) memory from the Cache Size Manager depending on the weight.

9.5 Content Manager

The Content Manager contains information on FirstSpirit database connections and accesses.

Attributes:

ExecutedQueries: number of the database SQL statements executed within the last minute.



9.6 Discovery Manager

The Discovery Manager finds other servers within the current network by transmitting and responding to broadcast messages. This entry is used to collect and prepare information about these servers. This information can also be viewed using ServerMonitoring in the "FirstSpirit – Control – Network" area (see Chapter 8.6.2.7 page 484).

9.6.1 Attributes

`NewestServer`: the time stamp of the most recently identified server.

`OldestServer`: the time stamp of the oldest identified server.

`ServerCount`: total number of servers in this network.

9.6.2 Operations

`void updateServers()`: calling this method updates the server information. This action can also be executed from FirstSpirit ServerMonitoring (see Chapter 8.6.2.7).

9.6.3 Attributes (server-based)

`Hostname`: host name of the FirstSpirit server. This value is defined using the `HOST` parameter in the `fs-server.conf` configuration file (see Chapter 4.3.1.1).

`HttpPort`: HTTP port of the FirstSpirit server. This value is defined using the `HTTP_PORT` parameter in the `fs-server.conf` configuration file (see Chapter 4.3.1.1).

`InetAddress`: FirstSpirit server address information.

`LicenseId`: FirstSpirit license ID (for information on FirstSpirit licensing, see Chapter 8.6.1.2).

`LicenseType`: FirstSpirit license type (for information on FirstSpirit licensing, see Chapter 8.6.1.2).

`Modules`: FirstSpirit server module information.

`SocketPort`: FirstSpirit server socket port. This value is defined using the `SOCKET_PORT`



parameter in the `fs-server.conf` configuration file (see Chapter 4.3.1.1).

Timestamp: time stamp of the server information shown.

Url: URL for FirstSpirit server start page. This value is defined using the `URL` parameter in the `fs-server.conf` configuration file (see Chapter 4.3.1.1).

User: user to which the FirstSpirit license was issued.

Version: FirstSpirit Server version.

9.7 Event Manager

The Event Manager provides information on fired and sent events. Many actions initiated within the FirstSpirit environment result in certain events. For instance, loading a project can cause corresponding references to be loaded via the Reference Manager. The Event Manager provides different managers (e.g. Reference Manager and Search Manager) for the interface.

Additional actions that fire events are:

- Create, edit, save and delete elements
- Load, change, deactivate and delete projects
- Create, edit, save and delete database content

The Event Manager determines the processing speed and load on the FirstSpirit server, particularly in multiuser environments.

Attributes:

EventListeners: number of server and client-side event listeners currently registered on the server.

EventQueues: the event queues are the server-side equivalent of the client-side event listeners. Since the server cannot place the events directly on the client, these events are placed in a queue until they can be retrieved (see also `EventsQueued`). This value provides indirect information on how many clients are connected to the server.

EventsFired: events fired in the last 60 seconds.

EventsQueued: number of events that have been fired and not yet processed (see also `EventQueues`) (within the last 60 seconds).



`EventsSent`: number of actual events sent (to an event listener) within the last 60 seconds.

`LastFiredEvents`: the last events fired (string array).

9.8 ExecutionManager

In FirstSpirit, a large number of actions are carried out concurrently in the background. This includes, for instance, updating reference graphs, processing client calls or even indexing documents. The ExecutionManager handles execution of these actions.

9.8.1 Classification of thread queues

The ExecutionManager manages a large number of differently classified thread queues. A thread queue contains the different tasks (client calls, for instance) up until they are executed. Thread queues can be configured based on their classification.

`ThreadQueue.SERIAL`: In the `SERIAL` classified queues, tasks are prepared that cannot be run concurrently. Since only one task can be carried out at a time, limiting the queue through configuration is not necessary.

`ThreadQueue.LOW`: of the memory or processor-intensive tasks prepared in the the queues classified as `LOW`, only a few can be executed simultaneously (default value is 2). An example of this is the indexing of documents. The number of tasks that may be run at the same time can be configured using the `maxRunning` parameter (see Chapter 4.3.1.5 page 43).

`ThreadQueue.DEFAULT`: `DEFAULT` is the default queue classification. All tasks are prepared in a queue classified as `DEFAULT` if they are not assigned to a different queue classification. The number of tasks that may be run at the same time can be configured using the `maxRunning` parameter (see Chapter 4.3.1.5 page 43).

`ThreadQueue.HIGH`: the queue classified as `HIGH` is used for all tasks that are to be run immediately. An example of this would be high-priority client calls (e.g. ping). Configuration is also unnecessary in this case.

`ThreadQueue.BOUNDED`: tasks in a queue classified as `BOUNDED` prepare the server calls of clients only. This queue can be limited in two ways. The number of active tasks and the capacity of the queue can be limited (for configuration, see Chapter 4.3.1.5 page 43). If the maximum capacity of the queue is reached, additional client calls from the server are temporarily rejected before another attempt is



initiated (rejection strategy).

9.8.2 Processing within the ExecutionManager

The following diagram illustrates how tasks are executed in the ExecutionManager. The values described here are available in the JMX console (see next page), and some are available in FirstSpirit ServerMonitoring (see Chapter 8 page 449):

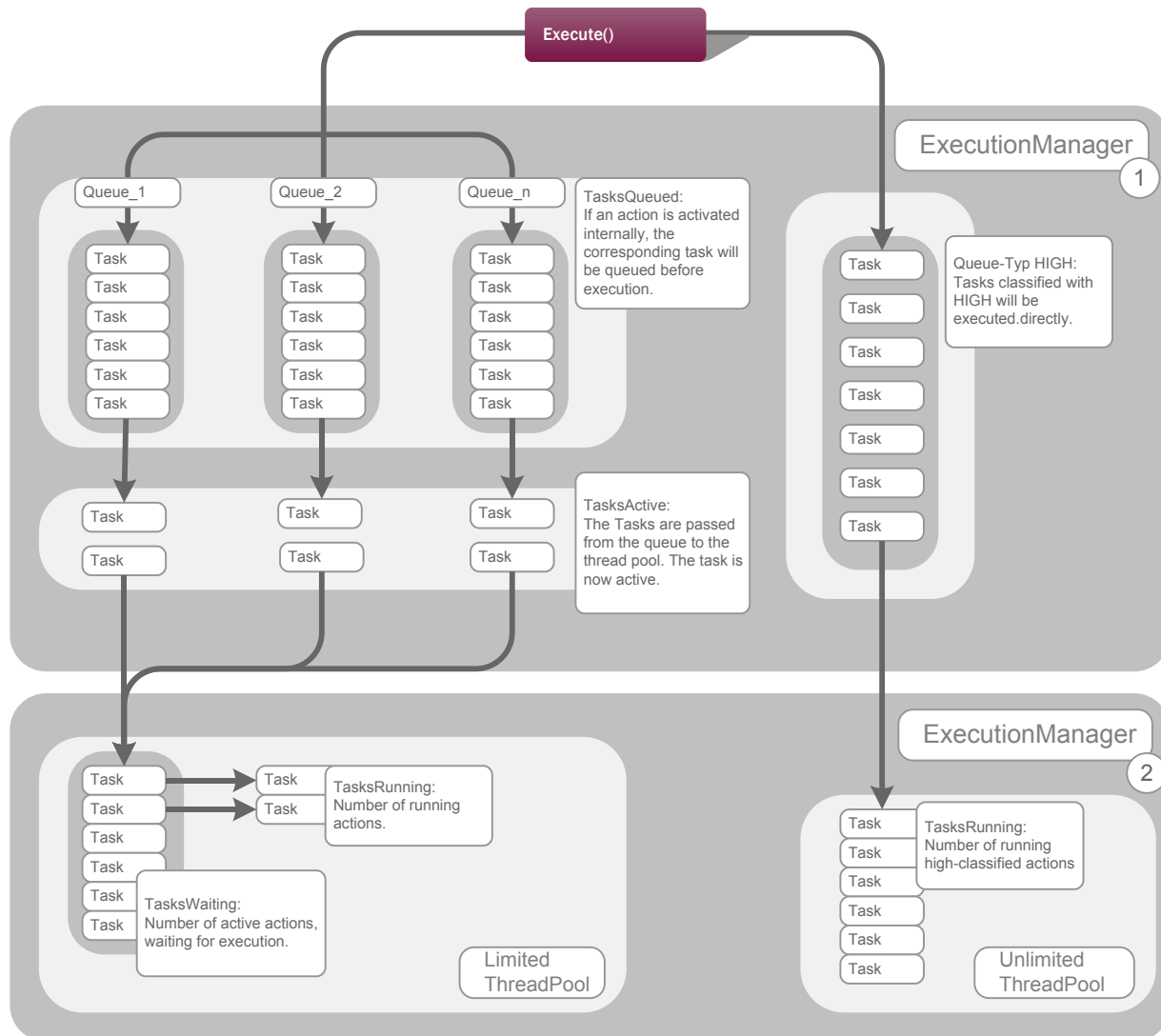


Figure 9-6: Execution of tasks using the ExecutionManager



9.8.3 Attributes

ExecutionRate: number of tasks executed within the last 60 seconds.

LastStartedTasks: shows the last tasks started (string array).

ScheduledTasks: in addition to the events triggered by user actions, there are also periodically scheduled internal actions. For instance, once every minute a check is made to determine if a session is still valid or if it has already finished. The value reflects the number of all regular pending actions.

TasksActive: a queued internal action (see "TasksQueued") is passed from the queue to the second processing level of the ExecutionManager (see Figure 9-6). The relevant task is then active, but not yet executed; it enters into the (limited) thread pool. The number of possible active tasks that enter the thread pool from the queue can be limited using the `maxRunning` parameter (see Chapter 4.3.1.5 page 43).

TasksQueued: if an action is initiated internally, such as indexing, the corresponding task is first queued. The value shown here provides the number of currently queued actions (see Figure 9-6).

TasksEscaped: number of tasks executed outside of their actual queue because the queue is full.

Background: the ExecutionManager queues use a rejection strategy. If a queue is "bounded", which means it is limited in size, tasks above this limit are passed to the rejection strategy. The called thread places its task in the queue and executes the oldest entry itself. These tasks are called "escaped" because they are not executed through their actual queue. The "overloaded" queue handles what are known as the called threads, and the "active" tasks of the queue exceed "maxRunning". The queue capacity can be defined in the `fs-server.conf` configuration file (see Chapter 4.3.1.5 page 43).

TasksRunning: number of currently running actions (for instance, an indexing job or an event) within the limited and unlimited thread pool (see Figure 9-6). The number of concurrently running tasks (via the limited thread pool) can be limited using the `maxSize` parameter (see Chapter 4.3.1.4 page 41).



TasksWaiting: number of actions already transferred from the corresponding queue, but still waiting for execution in the thread pool (see Figure 9-6).

TotalExecuted: total number of executed actions.

ExecutionQueues: the ExecutionQueues entry provides an overview of all of the different managers' queues. The current queues can be viewed by double-clicking on an entry or using the left-hand navigation (see Figure 9-7).

```
Queue[2:SearchManager.dataIndexing,DEFAULT] - a:0 q:0 m:24 c:104
Queue[1:SearchManager.mediaIndexing,LOW] - a:0 q:0 m:2 c:130
Queue[3:SearchManager.searching,DEFAULT] - a:0 q:0 m:24 c:104
Queue[4:MediaManager,LOW] - a:0 q:0 m:2 c:130
```

Figure 9-7: ExecutionManager – Queues

Each entry contains the following information:

Name of queue: the name is composed of the following (see Figure 9-7):

- The queue ID
- The name of the manager using the queue
- The project ID (This value is optional. Some managers, such as the Reference Manager, generate a queue for each project.)
- The queue class. The possible classes are `LOW`, `HIGH SERIAL`, `BOUNDED`, `DEFAULT` (see Chapter 9.8.1 page 505). Some queue classes can be configured using the parameters in the `fs-server.conf` configuration file (see Chapter 4.3.1.5, starting on page 43).

Number of active threads: number of active threads [`a: number`] passed from the queue to the thread pool (see "TasksActive").

Number of queued tasks: number of queued tasks [`q: number`] waiting in the queue for their execution (see "TasksQueued").

Number of tasks that can run concurrently: number of tasks that are permitted to run simultaneously, [`m: number`] (see "maxRunning").



Maximum number of queued tasks: number of tasks `[c: number]` that are permitted to be queued within a "bounded" queue (see "queueCapacity" in Chapter 4.3.1.5 page 43)

Number of running tasks: number of tasks currently running `[r: number]` (such as an indexing job or an event) in the limited and unlimited thread pool (see "TasksRunning").

Number of waiting threads: number of queued threads `[w: number]` passed from the queue to the thread pool that have not yet been executed (see "TasksWaiting").

9.9 Lock Manager

In FirstSpirit, objects need to be blocked from access by other users for certain actions, such as changing project properties or editing a page. To do this, the user automatically (when changing project properties) or manually (when editing a page) sets a lock on this object. The Lock Manager manages all currently locked objects and provides information on different lock types.

Attributes:

LockedObjects: number of all objects (projects, elements, packages, schedule entries) that are currently being edited and have been locked by a user.

ObjectsLocked: number of all objects (projects, elements, packages, schedule entries) that were locked by a user within the last 60 seconds.

LockedPackages: number of all packages that are currently being edited and have been locked by a user. This value is only populated when the license-dependent "CorporateContent" functionality is in use.

LockedProjects: number of all projects that are currently being edited and have been locked by a user (for an example, see Chapter 7.3.13 page 278).

LockedScheduleEntries: number of all schedule entries that are currently being edited and have been locked by a user (for an example, see Chapter 7.5.4 page 380).

LockedStoreElements: number of all elements from the FirstSpirit page store that are currently being edited and have been locked by a user.

LockedDatasets: number of entities (database entities) that are currently being edited and have been locked by a user.



9.10 Media Manager

The FirstSpirit media store contains thumbnails for viewing. Media Manager manages this thumbnail information for display and calculation when adding a new medium and is also responsible for calculating resolutions, when requested.

Attribute:

ImagesScaled: number of scaled media (including, for instance, all recently calculated thumbnails).

9.11 Module Manager

FirstSpirit supports the integration of third-party components. The key component of this integration is the module, which in turn consists of one or more components. Each component uses resources (shared, web local – e.g. classes, Jar archive, properties files), covers a certain scope, has a type (e.g. project application, web application) and certain properties. All of a module's files are compressed in a zip file with the .fsm extension and can then be installed on the FirstSpirit server. The Module Manager is responsible for all functions related to managing modules and components on the FirstSpirit server. The MBean Module Manager provides a list of all modules (fsm files) loaded on the server.

For information on development, class loading and installation of modules, see the FirstSpirit Manual for Developers (Components) (German only).

9.12 NIO Socket Server

Communication between FirstSpirit servers and clients is handled via the NIO Socket Server. The associated MBean shows the manager's current measured values.

Attributes:

Host: socket host; the host name for the Socket_Port bind address used to limit the FirstSpirit server to one IP address when required. If there is no value here, the server binds to all host IP addresses. This value is defined using the `SOCKET_HOST` parameter in the `fs-server.conf` configuration file (see Chapter 4.3.1.1).

Port: FirstSpirit server sock port. This value is defined using the `SOCKET_PORT` parameter in the `fs-server.conf` configuration file (see Chapter 4.3.1.1).



ConnectionCount: number of currently open sockets for communication between the FirstSpirit server and FirstSpirit clients.

ConnectionCreationRate: number of socket connections recently opened within the last 60 seconds.

ConnectionGroupCount: number of FirstSpirit clients (SiteArchitect, ContentCreator, web applications) currently connected to the server. (Not the same as the number of open sockets.)

DanglingCallCount: The FirstSpirit server processes all client requests using the following strategy. The FirstSpirit server responds immediately to short requests. In the case of requests that require a response exceeding 30 seconds, the server response is no longer returned immediately, but is instead processed by the server and then provided for retrieval by the client. The DanglingCallCount attribute returns the number of requests that were processed by the server and were prepared for retrieval.

LongDurationCallCount: number of long duration requests (>30 sec.) that have not yet been completely processed by the FirstSpirit server and have therefore not been prepared for retrieval by the client.

ResultWorkerCount: number of client requests currently being processed.

9.13 Preview Manager

FirstSpirit can be used to request a preview any time of the content currently being edited. A preview can be requested within the page store at the page level or within the site store at the site reference level. The Preview Manager manages all previews currently requested and provides the information for them.

Attributes:

AveragePreviewCount: number of pages or page references for which a new preview was calculated within the last 60 seconds.

RunningPreviews: number of previews requested that are currently being calculated.





Requesting a preview is a very CPU and time intensive task. FirstSpirit therefore offers the ability to move the calculation of a preview to external systems. In this case, the Preview Manager will not be able to analyze any information related to the preview.

9.14 Project Manager

The Project Manager primarily provides information on the exact number of objects (grouped by object types and stores) of a project. Each project has its own MBean, which is displayed under the "ProjectManager" folder. This information is not updated dynamically. The calculation is started once the `calculateCounters()` method is called. This updates the desired information in the Attributes area while at the same time setting the value for the `_Status` attribute to `CALCULATED`.

9.14.1 Attributes

`All`: number of all the project's objects.

`All_ContentStore`: number of all objects in the project data store.

`All_MediaStore`: number of all objects in the project media store.

`All_PageStore`: number of all objects in the project page store.

`All_SiteStore`: number of all objects in the site store.

`All_TemplateStore`: number of all objects in the template store.

`ContentStoreDefinitions`: number of data sources in the project data store.

`ContentStoreFolders`: number of folders in the project data store.

`MediaStoreFolders`: number of folders in the project media store.

`MediaStoreMedia`: number of media (files, images) in the project media store.

`PageStoreBodyNodes`: number of content areas in the project page store.

`PageStoreSections`: number of sections in the project page store.



`PageStoreFolders`: number of folders in the project page store.

`PageStorePages`: number of pages in the project page store.

`SiteStoreDocumentGroups`: number of document groups in the project site store.

`SiteStoreFolders`: number of menu levels in the project site store.

`SiteStorePageReferences`: number of page references in the project site store.

`TemplateStoreFormatTemplates`: number of format templates in the project template store.

`TemplateStoreLinkTemplates`: number of link templates in the project template store.

`TemplateStorePageTemplates`: number of page templates in the project template store.

`TemplateStoreTableTemplates`: number of table templates in the project template store.

`TemplateStoreWorkflows`: number of workflows in the project template store.

`_Status`: status of the calculation `NOT_CALCULATED` | `CALCULATED`. The calculation is started by executing the `calculateCounters()` method and then the value `CALCULATED` is set; executing the `resetCounters()` method resets it to the value `NOT_CALCULATED`.

9.14.2 Operations

`void resetProjectStores()`: when executing the method, the cached, server-side Access API stores are reset. This may be practical, for instance, when adding or removing resolutions or languages in a project and these changes are not yet visible in the project. Note: this method should not be executed while the server is running.

`void resetCounters()`: when executing the method, the number of attribute values previously calculated is reset to 0. After execution, the value of the `_STATUS` attribute changes to `NOT_CALCULATED`.

`void calculateCounters()`: the number of project objects provided as attribute values in the individual MBeans need to be calculated in advance. To do this, the `calculateCounters()` method needs to be executed. After execution, the value of the `_STATUS` attribute changes to `CALCULATED`.



9.15 Reference Manager

Essential FirstSpirit functions are based on a project's reference graph. The reference graph is used to identify dependencies of objects within complex projects. The Reference Manager is responsible for calculating and supplying these references. The calculation of references is usually automated and done while the project is being worked on. In some cases, however, manual recalculation of the references may be necessary. The recalculation of references is asynchronous, i.e. servers and projects can still be used while the reference calculation is running. However, this is not recommended, since some operations that analyze the reference graph may be relying on incomplete information. Therefore, objects used in a project are deleted if the incoming references have not yet been calculated in their entirety.

9.15.1 Attributes

ActiveCalculated: current number of all project references being recalculated on the FirstSpirit server.

ActiveDuration: current duration of the actively running recalculation of project references on the FirstSpirit server. This information is also shown in ServerMonitoring under "Overview – Activities" (see Chapter 8.1.2 page 451).

ActiveQueued: current number of project references waiting to be calculated on the FirstSpirit server.

ActiveRecalculations: current number of project references that are being recalculated now.

ActiveRepaired: current number of repaired project references on the FirstSpirit server. This information is also shown in ServerMonitoring under "Overview – Activities" (see Chapter 8.1.2 page 451).

CalculationRate: number of references that were recalculated within the last 60 seconds (references per minute).



9.15.2 Operations

`void updateProjectReferences()`: this method starts recalculation of all obsolete project references on the FirstSpirit server.

`void startRecalculation(projectID)`: this method starts recalculation of the most recent revision of project references for a particular project. The JConsole is used to pass the project ID for which a recalculation of the references is to be started.

The recalculation of project references can also be monitored in ServerMonitoring under "Overview – Activities" (see Chapter 8.1.2 page 451).

9.16 Registry Manager

Since FirstSpirit version 5.0, central registries are used to store (in a NoSQL database) certain internal project data, such as predefined URLs or even GIDs (global identifiers), which are needed for FirstSpirit content transport. Each project can have multiple registries. The Registry Manager manages the registries.

Operations:

`void deleteRegistry (projectID, registry name)`: this method deletes a project's registry and all the data contained therein. To do this, the project ID and the name of the database must be specified.

9.17 Repository Manager

FirstSpirit uses content repositories to archive and maintain version histories of project data. The content repository is a central location where the required file structures (media pages, templates, etc.) are managed by the Content Management System. Each FirstSpirit project has its own self-contained content repository; for each repository there is an MBean in the JMX console "Repository" folder. Display is dynamic. This means that the entry is loaded in the JMX console only after the corresponding project has been opened by at least one user. The information is then refreshed dynamically at runtime.

BackendClass: class that implements the corresponding repository (currently only Berkeley DB is supported).



`BasePath`: full path to the project's repository.

`LastClosedRevision`: number of last revision released for this project.

`RevisionCounter`: total number of revisions for this project.

`UncommittedRevisions`: total number of revisions currently being edited (not yet released).

9.18 Search Manager

The Search Manager is responsible for indexing and searching for elements from FirstSpirit stores (e.g. page, sections and media files). The elements are written to or removed from the search index when they are created, edited or deleted.

Full text indexing of a large amount of data (such as several extensive documents from the FirstSpirit media store) is time and CPU-intensive. The amount of documents indexed simultaneously can be controlled globally via the settings in the `fs-server.conf` configuration file:

- `ThreadQueue.DEFAULT.maxRunning`: for the content and variables of store element objects.
- `ThreadQueue.LOW.maxRunning`: for documents from the media store (e.g. PDF files).

9.18.1 Attributes

`IndexingStarted`: number of elements created, changed or deleted in the last 60 seconds that have to be indexed using the Search Manager. These actions are queued and processed as low priority.

`DocumentsIndexed`: number of elements that were re-indexed within the last 60 seconds (indexing has been completed).

`SearchStepsExecuted`: every search consists of a number of different search steps (3 to 5 steps per search). The value here is the number of search steps executed within the last 60 seconds.





Full text indexing of the media store: documents from the FirstSpirit media store are also indexed so that a full text search is possible within the documents. For technical reasons, not all document types are supported. (Complete) indexing is therefore not always possible.

9.18.2 Operations

`void pause()`: this method interrupts the indexing of project content. Indexing can then be resumed at a later time by executing the `resume()` method. These two methods can be set to interrupt indexing briefly while importing a large volume of data into a FirstSpirit project, for instance.

`void resume()`: this method resumes indexing of project content after it was previously paused (see `pause()`).

`void rebuildIndex(projectID)`: this method initiates manual updating of the search index for a project. The corresponding project ID is passed. When executing the method, the search index is not completely rebuilt. Information is just added to the existing index. If the search index has become corrupted, the method `clearIndex(projectID)` should be executed first.

`boolean clearIndex(projectID)`: using this method, the search index of a project can be deleted (e.g. because it has become corrupted). A new empty index is then created for the project. To re-index the project, the method `rebuildIndex(projectID)` should then be executed.

9.19 Server Action Manager

Many CPU-intensive actions, such as determining the dependencies of objects during release, are shifted from the FirstSpirit clients to the server. The Server Action Manager manages these server actions and also ensures that the actions are cleared periodically after processing.

9.19.1 Attributes

ActionCount: number of all server actions currently being executed on the FirstSpirit server. Depending on the particular implementation, a situation may arise where after a server action is executed, the server action handle is not removed. In this case,



the remaining server action handles also increase the action count and are removed periodically after a waiting time (default value is 5 min.).

Actions: shows all FirstSpirit server actions (string array).

9.19.2 Operations

`java.lang.String getResult(actionID)`: the method returns the result after the server action is completed, e.g. `result=true` for release server actions. This is possible if the server action has already been completed, but the associated server action handle still exists. In the case of internal FirstSpirit server actions, the server action handle is automatically removed after the action has been completed.

`java.lang.String cancelAction (actionID)`: when executing this method, the server action is canceled using the assigned action ID. The result of the action is returned.

`boolean isActionRunning(actionID)`: the method returns whether the server action is still executed on the server using the assigned action ID or not.

9.20 Server Manager

The Server Manager manages general data and methods for controlling and configuring the FirstSpirit server.

9.20.1 Attributes

IgnoreStopServerRequests: shows whether the FirstSpirit server control via the `ignoreStopServerRequests(pwd)` method is disabled (true) or not (false).

Version: shows the version of FirstSpirit Server based on this format: `major.minor.build.revision`

9.20.2 Operations

`java.lang.String dumpThreads()`: when executing the method, a thread dump of the server is generated and returned.



`boolean logThreadDump()`: when executing the method, a thread dump of the server is generated and written to the server log file (fs-server.log) with the INFO log level.

`boolean enableVerboseMemoryLogging()`: the method reroutes all garbage collector log outputs (covering the Java VM memory load) (see 4.3.2.1) to the console (command line/shell) started by FirstSpirit.

`boolean disableVerboseMemoryLogging()`: the method disables rerouting of the garbage collector log output to the console (see `enableVerboseMemoryLogging()`) and resets the status originally defined for the log output (see Chapter 4.3.2.4 page 81).

`void clearLoggerCache()`: this method resets the cache for log instances.

`void restartManager(manager)` this method restarts a manager. The name of the desired manager must be provided to do this.

`void ignoreStopServerRequests(pwd)`: this method prevents the server from being stopped by the conventional server control (see Chapter 3); it has been introduced for the internal testing infrastructure.

`void resetIgnoreStopServerRequests(pwd)` this method restores the conventional status of the FirstSpirit server control (see `ignoreStopServerRequests(pwd)`).

9.21 Service Manager

The Service Manager manages and controls the FirstSpirit system services. A service is a server component that can be activated via a public interface made of input components or scripts. (Examples include spell checking or the CMS_INPUT_PERMISSION permissions input component service.) The services can also be configured and controlled through FirstSpirit ServerMonitoring (for configuration, see Chapter 8.6.1.7, and for control, see Chapter 8.6.2.4) or FirstSpirit ServerManager (see Chapter 7.3.11 page 265).

9.21.1 Attributes

`Services`: shows all FirstSpirit Server services (string array).



9.21.2 Operations

`java.lang.String stopService(serviceName)`: stops the service using the assigned name.

`java.lang.String startService(serviceName)`: starts the service using the assigned name.

`java.lang.String restartService(serviceName)`: restarts the service using the assigned name. Unlike simple start or stop functions, the service that is already started is first stopped and then automatically restarted.

`java.lang.String setAutostartOn(serviceName)`: enables automatic starting of the service with the assigned name. The service is now automatically started every time the server is restarted.

`java.lang.String setAutostartOff(serviceName)`: disables automatic starting of the service with the assigned name. When the server is restarted, the service then has to be restarted manually.

9.22 Session Manager

The Session Manager is responsible for managing the server sessions. When a user logs in, a new session is created on the server. The Session Manager regularly checks to see if a session is still valid or if it has already finished (see Chapter 9.8 page 505). An overview can also be requested in FirstSpirit ServerMonitoring (see Chapter 8.1.3 page 452).

9.22.1 Attributes

`ActiveProjectsCount`: number of projects with active sessions.

`ActiveUsersCount`: number of users with an active session.

`LicenseSessionCount`: number of sessions requiring a FirstSpirit license based on the FirstSpirit licensing terms and conditions. Temporary sessions, i.e. sessions created by the server, are not shown.

`MaxSessionCount`: maximum number of possible sessions based on the FirstSpirit license (see Chapter 8.6.1.2 page 471).



`SessionCount`: number of currently valid sessions. Temporary sessions, i.e. sessions created by the server, are not shown. Note: updates, e.g. user manually deleting cookies, are potentially displayed only after a timeout of 20 minutes.

`SessionCreationRate`: number of sessions recently opened within the last 60 seconds.

`TicketCount`: number of tickets on the FirstSpirit server.

9.22.2 Operations

`void dumpSessions()`: executing this method writes the session information to the server log file (fs-server.log).



10 Secure deployment via rsync and ssh

The combined application of the external service programs `rsync`³⁷ and `ssh`³⁸ 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 Chapters 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`.

³⁷ `rsync`: <http://rsync.samba.org/>

³⁸ `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 <code>/etc/ssh_config_file</code> ?	Yes
Overwrite existing <code>/etc/sshd_config_file</code> ?	Yes
Create local user <code>sshd_server</code> ?	Yes
Should privilege separation be used?	No
Do you want install <code>sshd</code> as service?	Yes
Value for environment variable <code>CYGWIN</code> =?	ntsec tty
8. 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  
mkgroup -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 (fs5) or use “su - fs5” 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 “fs5” 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 -vcrzt -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 Cygwin³⁹ 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 account.

³⁹ <http://cygwin.com/>



"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 -vrtz -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 Chapter 7.5.4 page 380).
2. Add the "Generate project" action to the schedule entry (see Chapter 7.5.10.2 page 406).
3. Add the "Execute script" action below the previous action (see Chapter 7.5.9.4 page 395).
4. The "ssh/rsync deployment script" is part of a zip archive (admi5x_files.zip) for the *Documentation for Administrators* and can be downloaded from the FirstSpirit Online Documentation (ODFS).
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 the entered configuration should be tested and, if necessary, the displayed error message log should be checked. 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 Chapters 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 personalization system. If FirstSpirit DynamicPersonalization⁴⁰ is used as the personalization system (not mandatory), a very close relation can be established (see Chapter 11.2.3 page 537).

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 user permissions are used in a project.

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

⁴⁰ See FirstSpirit Personalization documentation



the “View” operation. In this case, a 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 site⁴¹:

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 login option is available⁴²:

- SSO: If an SSO module is used, the transition from editor to user takes place transparently without a password request (login module: FS SSO).

11.1.1 Define user permissions

User permission assignment is always based on groups, since experience has shown that management 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 hierarchical structure – i.e. a group can contain several sub-groups.

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 a special input component

⁴¹ See FirstSpirit DynamicDatabaseAccess documentation

⁴² See FirstSpirit Personalization documentation



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 DynamicPersonalization module.

Another logical consequence is the need to generate an effect from the permission evaluation – i.e. a reaction. Sometimes the reaction occurs within the scope of personalization.



This involves parts of a page or even the complete page being protected by special tags (FirstSpirit Personalization tags⁴³). 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 539). 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 541). For more information about “Protection of personalized 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 539). 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 Chapter 11.2.3 page 537 ff.

11.1.3 Assigning user permissions

In FirstSpirit, the user permission assignment and check take place via two components:

- Editor (permission component) - (see Chapter 11.1.3.1 page 531)
- Service (PermissionService) - (see Chapter 11.1.3.2 page 532)

These both components are part of the system module which is already included when installing a FirstSpirit Server and need not to be installed separately.

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 form) parameterised with a unique group

⁴³ See FirstSpirit Personalization documentation



document and cooperates with the appropriate service which loads and provides the group definitions from the server (see Chapter 11.1.3.2).

See the “FirstSpirit Online Documentation” 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 ServerMonitoring (for configuration of the file `fs-server.conf` via FirstSpirit ServerMonitoring see Chapter 11.4.2 „Activating the permission service“ page 547).

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 is used here.

The relations and the dataflows between the components are essential for understanding interaction between the individual components.



11.2.2 Overview

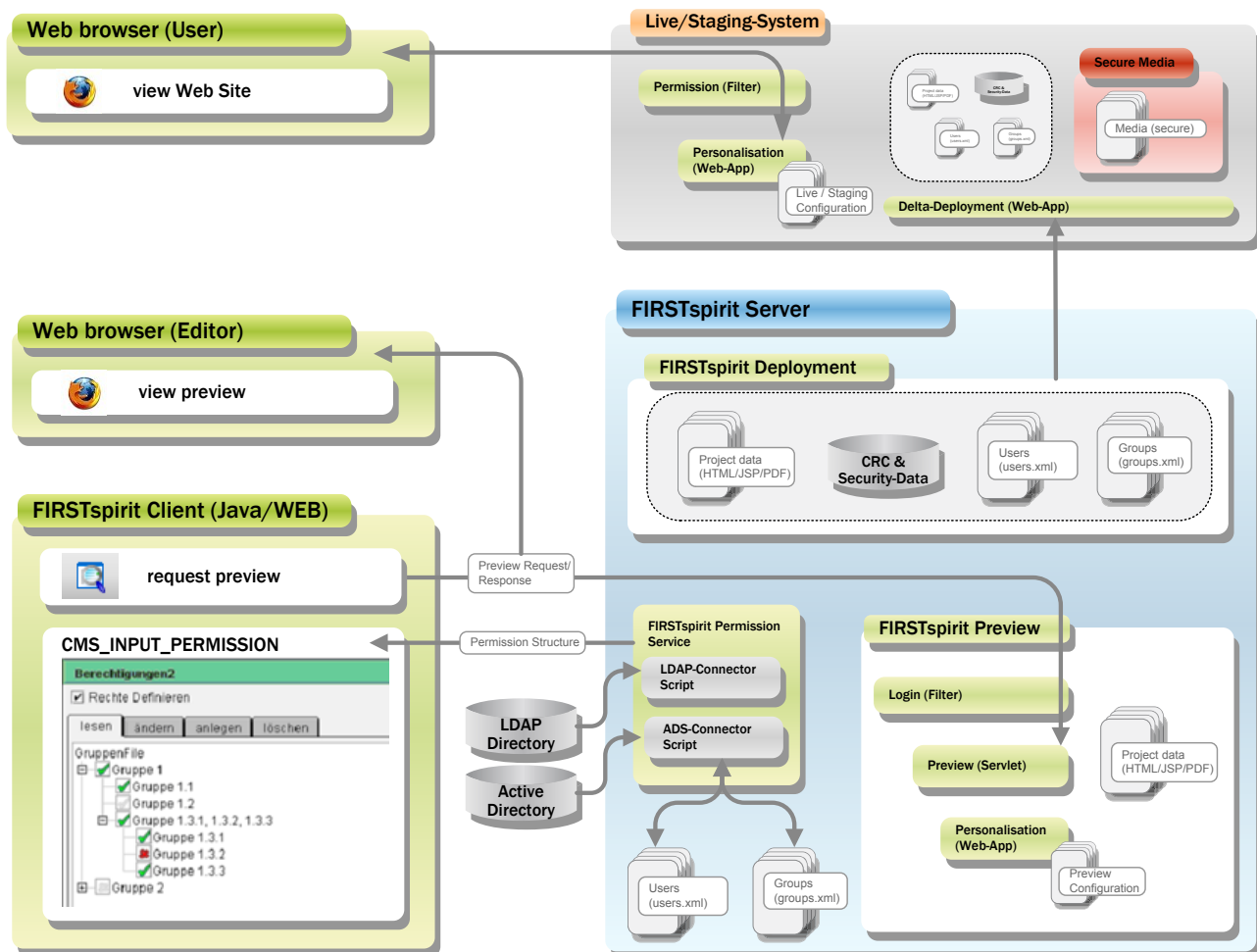


Figure 11-1: Live system overview

FirstSpirit Clients: Communicate exclusively with the FirstSpirit server and never directly with, e.g., an LDAP server or database (for further information see Chapter 11.2.2.1 page 534).

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 Chapter 11.2.2.1 page 534).

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 Chapter 11.2.2.2 page 535).

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 Chapter 11.2.2.3 page 536).

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 Chapter 11.4.2 „Activating the permission service“ page 547).

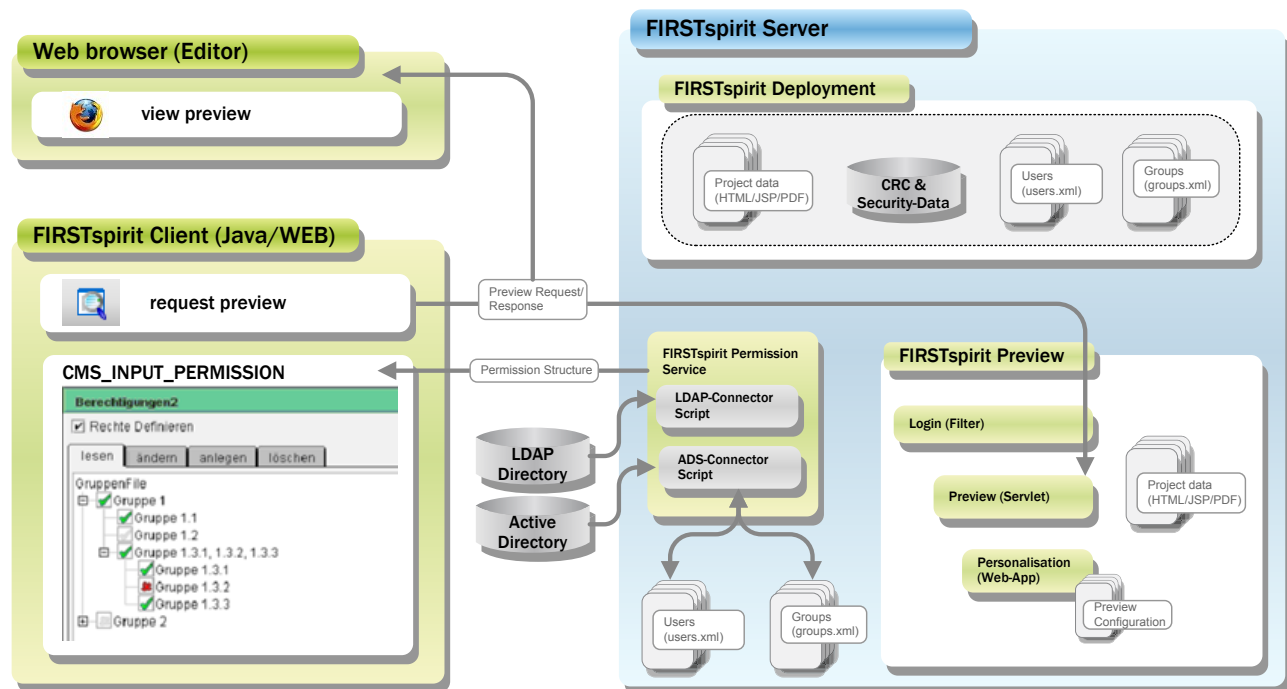


Figure 11-2: Live system extract



The group and user configurations managed by the “permission service” are configured in the service configuration file (“`service.ini`”) (see Chapter 11.4.3.1 „Structure of the service configuration file `service.ini`“ page 548). The respective XML files (see FirstSpirit Manual for Developers) can be generated manually (via FirstSpirit ServerMonitoring, see Chapter 8.6.1.7 page 475), or automatically via a connector script based on an exiting user/group management 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 Chapter 11.6 “Application in the project” it is possible to use the permission definition as “target permission” for the personalization (see Chapter 11.3.1 page 538), 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 540).

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 541). 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 extensions in the preview.

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 540). 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 personalization) 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 Chapter 11.2.2.1 page 534)



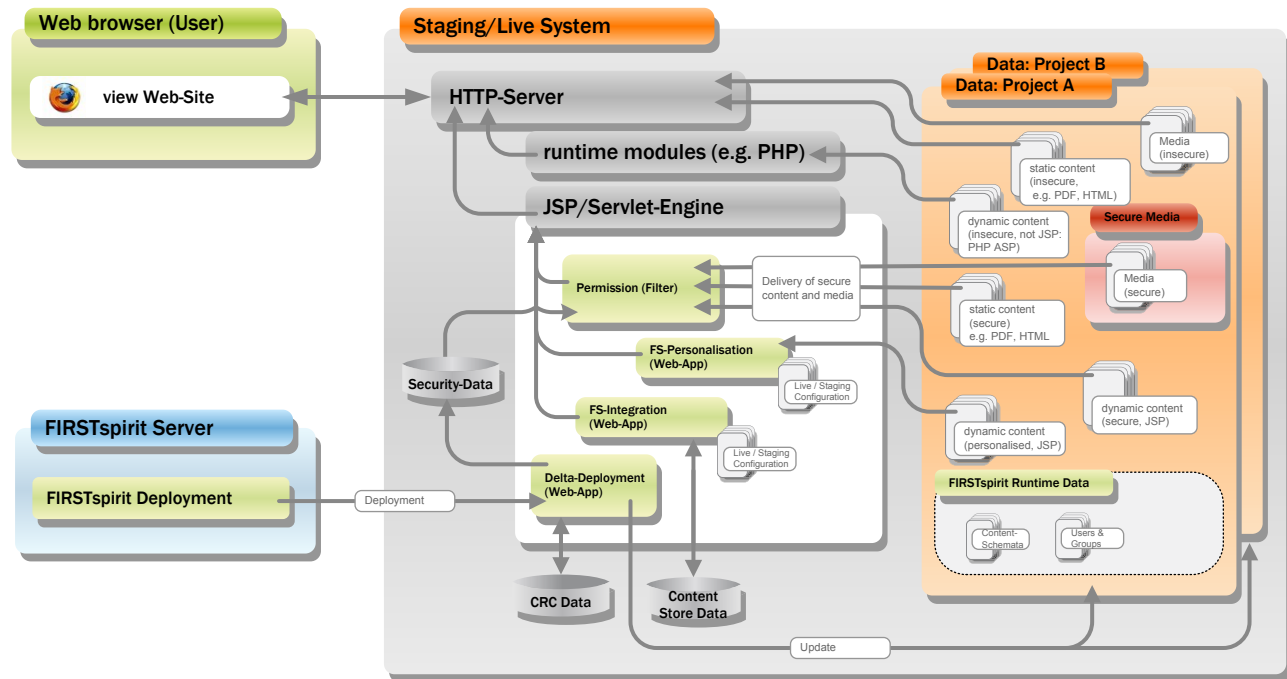


Figure 11-3: Live and staging system

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 539).

11.2.3 Authorization checks using FirstSpirit

11.2.3.1 Quick start

Quick start provides an overview of the steps required and the order in which they should take place in order to enable authorization checks in FirstSpirit:

1. Starting the permission service: the service is a system module component and is present on the server when first installed, so it does not need to be installed separately. However, the service contains services that may need to be started if they are not already started automatically. Starting and stopping services is handled either via FirstSpirit ServerMonitoring (see Chapter 8.6.2.4 page 482) or via ServerManager (see Chapter 7.3.11 page 265).
2. Adapting the permission service's system configuration files (`service.ini`, `groups.xml` and `users.xml`) (see Chapter 11.4.3 page 547)
3. Creating a metadata template (see the FirstSpirit Manual for Developers (Basics))
`CMS_INPUT_PERMISSION` and `secure media`
4. Configuring web components using ServerManager (see Chapter 7.4.18 page 343).
5. Configuring permissions using ServerManager (see Chapter 11.4.5 page 553)
 - Input component information for secure media
 - Input component information for checking permissions
6. Configuring the Multi-Access Control filter (see Chapter 11.3.2.2 page 540)

See also the FirstSpirit security module documentation.

The following Chapters describe in detail the implementation scenarios and individual configuration options.



11.3 Protection of personalized project content in FirstSpirit

FirstSpirit has different concepts to protect project content from unauthorised access:

- Personalization of content (see chapter 11.3.1 page 538)
- Checking the access rights to an object via the Access Control database (see chapter 11.3.2 page 539)
- Access protection for media within FirstSpirit preview ("secure media") (see chapter 11.3.3 page 541)

11.3.1 Personalization

The module FirstSpirit DynamicPersonalization⁴⁴ consists of an easy-to-use Java TAG library for personalizing 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 personalization 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 personalization refers.

Example: 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 personalization module can be established as follows:

1. The generation of the group definition file for the permission component and the "group module" of the personalization evaluate the same external data sources (e.g.: LDAP server or AD server). In this case, data sovereignty is completely external.

⁴⁴ Module FirstSpirit DynamicPersonalization



2. The permission component generates a group definition file and the personalization module uses this data or refers to this file. The permission component can generate this file by requesting an LDAP server. The difference is that the personalization module does not request the LDAP server, but uses the data basis generated by the permission component.

While the “group module” of the personalization 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 Chapter 11.4.1 page 547).

Permission checks can only be forced on JSP pages through use of personalization in FirstSpirit. The personalized 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 539).

The FirstSpirit Personalization module is an additional function which has to be purchased.

For further information see the FirstSpirit Personalization 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 FirstSpirit 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 concept of "Secure media" 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)⁴⁵.

Configuration takes place via FirstSpirit ServerManager (see chapter 7.4.16 page 339).

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 7.5.10.6.3 Page 419), 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

⁴⁵ FirstSpirit Security module



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 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 541)
- Within the Live systems via a filter (Multi-Access Control Filter) (see chapter 11.3.3.2 page 542).

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 339). 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 534). 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).

The preview servlet is available via the standard web application fs5preview. 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



The FirstSpirit security module introduced in an earlier version of FirstSpirit requires configuration changes (when in the "live" state), since secure access is no longer limited only to media and now includes all project content.

The secure media principle refers only to the project preview. If media are to be protected in the live state as well, more steps are necessary.

To protect secure media within the live system, the FirstSpirit security module must be installed and configuration of a separate filter adapted to the secure media directory is required (see Chapter 11.3.2.2 page 540)⁴⁶. In this case, the permissions are not monitored by the preview servlet, but are instead based on information from the Access Control Database (see Chapter 11.3.2.1 page 540).

For more information, see the FirstSpirit security module documentation.

11.3.4 Scope

All the mechanisms described 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 Personalization

It is also possible to use a different permission component or personalization module, but this would require adaptations.

⁴⁶ Refer to the module documentation



Basically, permission checks on the section layer (i.e. within a page) are only possible via the FirstSpirit personalization module. If independent objects (e.g. pages, media or PDF documents) are to be protected, the “secure media” mechanism (see chapter 11.3.3 page 541) or the “Checking of access rights via the Access Control Database” (see chapter 11.3.2 page 539) can be used in addition to personalization.

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 Chapter 11.3.5.2 page 543).
2. The stores also form a hierarchy which is often used for inheriting permission definitions (see Chapter 11.3.5.3 page 545).
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 which is located (in the permission definition component) below the root node. 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 below the root node:

- 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, Chapter 11.6.2.1 page 557).
3. The following three states are defined in the group tree:
 -  Allowed – green tick
 -  Forbidden = red cross
 -   like parent node = either grey tick or grey cross.

Click through the three states. If necessary, the grey icons in the subordinated 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, which is displayed in the client and used by the editor to define the permissions. The input component determines the group configuration defined for the actual project by communicating with a FirstSpirit server component. The server component adopts the group hierarchy structure (e.g. based on an LDAP tree) and transmits the data to the FirstSpirit client.

11.4.2 Activating the permission service

The permission component and permission service are part of a FirstSpirit system module. This module is included on each newly installed FirstSpirit server and needs not to be installed.

Starting and stopping the permission service is handled either by FirstSpirit ServerMonitoring (see Chapter 8.6.2.4 page 482) or by ServerManager (see Chapter 7.3.11 page 265).

11.4.3 Configuring the server component

The server component is configured in a `service.ini` service configuration file in which the general settings are configured and a number of group definition files are optionally specified in XML format (see Chapter 11.4.3.1). The service configuration directory for modules is under the FirstSpirit directory in the "conf/modules" subdirectory. The individual service configuration files are in a subdirectory named after the component (in this case: `System.PermissionService`). The permission component uses the `services.ini` file in the following directory:

`conf/modules/System.PermissionService`

Information needs to be added to the `service.ini` file so that it is possible to form a group hierarchy based on LDAP queries. If this does not appear to be possible due to static parameterization, a "generating BeanShell script" may need to be added as a parameter. A suitable initial LDAP context can be provided in this script.



11.4.3.1 Structure of the service configuration file `service.ini`

The `service.ini` file configures the permission service. The key task of the permission service is to provide group hierarchies to the respective input components. Different group hierarchies can be defined that can be identified using a unique name. This makes it possible to define a group hierarchy for each project or to use multiple group hierarchies in a project.

Group hierarchies can basically come from two different sources:

- 1) From an explicitly specified XML file (created manually or through external automation). These XML files can be manually created in FirstSpirit ServerMonitoring (see Chapter 8.6.1.7 page 475).
- 2) Using a script: the group hierarchies can be created automatically using 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 generated for caching the results, but it is generated from a script. The script is called by the service at defined intervals and can modify the XML file, if necessary. A typical, applicable case for a script is the creation of a group file from an LDAP server.

The following global parameters are included in the INI file:

`interval` = period in seconds in which the INI file is checked for changes.

`documents` = comma-separated list of symbolic names of available group hierarchies.

`NAME.path` = path to group XML file (see also Chapter 11.4.3.2 page 550).
`NAME` is a place holder for a value of `documents`

If the group hierarchy is to be generated using a script, a number of parameters are required:

`NAME.path` = path to group XML file (see also Chapter 11.4.3.2 page 550)

`NAME.users` = path to user XML file (see also Chapter 11.4.3.3 page 551)

`NAME.script` = path to the BeanShell script (e.g. for automatic generation of user and group files – `users.xml` and `group.xml`).

`NAME.script.interval` = interval in seconds in which the script is to be called.

(The above mentioned `NAME` . parameters are possible parameters for each `documents` entry.)



If a group hierarchy is to be created from an XML file only the parameter `NAME.path` is required.

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
```

In addition, parameters can be specified for the LDAP connection:

`NAME.ldap.URL` = LDAP server URL,
`NAME.ldap.userDN` = login for LDAP lookup,
`NAME.ldap.password` = password for LDAP lookup,
`NAME.ldap.version` = 2 (LDAP protocol version), and
`NAME.ldap.ssl` = 0|1 specifies whether the LDAP connection is established using SSL.

An LDAP context (`javax.naming.directory.*`) that is available to the script is generated from the LDAP parameters entered here.

Here is an example of 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
```



11.4.3.2 Groups XML file (groups.xml)

The `groups.xml` group file where groups are defined is available by default. It needs to be introduced to the permission service configuration file `service.ini` by using the `NAME.path` parameter (see Chapter 11.4.3.1 page 548) and can be edited in FirstSpirit ServerMonitoring (see Chapter 8.6.1.7 page 475, "File name" under "Configuring a service").

It is recommended that you create a file using a script to reduce the probability of errors.

The following is the default configuration of the `groups.xml` file:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<GROUPS name="GroupsFile" version="1">
    <GROUP id="2" name="Anonyme Besucher"/>
    <GROUP id="3" name="Registrierte Mitglieder"/>
    <GROUP id="4" name="Kunden"/>
</GROUPS>
```

The name of the group definition can be assigned in the `name` attribute for the root element of the `GROUPS` group definition. In addition, the optional `version` attribute can be specified to record the respective group definition version number.

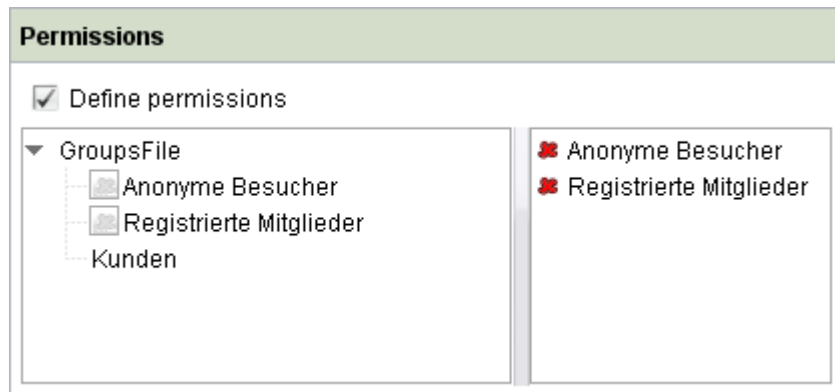
Within `GROUPS` any number of differently nested `GROUP` elements can be specified, where there is one group for each `GROUP` element. Each `GROUP` element requires the `name` attribute (name of group). The optional `id` attribute (group ID) can also be specified. This ID is required in order to be able to assign permissions to a group in an input component. If a group does not have an ID, it will be displayed in the input component without a node where permissions are set.

The following is an example of configuration without the `id` attribute for the "Customers" group:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
    <GROUPS name="GroupsFile" version="1">
        <GROUP id="2" name="Anonyme Besucher"/>
        <GROUP id="3" name="Registrierte Mitglieder"/>
        <GROUP name="Kunden"/>
    </GROUPS>
```



Display in SiteArchitect:



The `selectable` parameter is used to specify whether a group in the input component is to be selectable or not. The values 1 (selectable) or 0 (not selectable) are passed. If no ID was assigned using the `id` parameter, then entering `selectable="1"` is not permitted.

In the case of `GROUPS` and `GROUP` elements, the `timestamp` (creation time stamp) attribute can also be specified.

11.4.3.3 Users XML file (`users.xml`)

To define users, a separate XML file must be created. This can be done in FirstSpirit ServerMonitoring (see the "Create file" button under "Configuring a service" in Chapter 8.6.1.7 page 475). This user XML file needs to be introduced to the permission service configuration file `service.ini` by using the `NAME.users` parameter (see Chapter 11.4.3.1 page 548)

It is recommended that you create a file using a script to reduce the probability of errors.



The following is an example of how to configure a user file:

```
<?xml version="1.0" encoding="UTF-8"?>
<USERS>
  <USER login="visitor_1"
        realname="Besucher 1"
        password="password_visitor_1"
        active="1"
        groups="2" />

  <USER login="member_1"
        realname="Mitglied 1"
        password="password_member_1"
        active="1"
        groups="3" />

  <USER login="client_1"
        realname="Kunde 1"
        password="password_client_1"
        active="1"
        groups="3,4" />
</USERS>
```

The `timestamp` (user definition time stamp) attribute can be passed to the root element of the `USERS` user definition.

Within `USERS` any number of `USER` elements can be specified, where there is one user for each `USER` element. Each `USER` element requires the following attributes:

- `login` (user login),
- `realname` (full user name),
- `password` (user password),
- `active` (active/inactive user, possible values: 1 and 0) and,
- `groups` (comma-separated list of group IDs to which the user is associated).

In addition, the optional attribute `timestamp` can be specified.

11.4.4 Configuration for the deployment

The required files are deployed via a deployment servlet (see Chapter 7.5.10.6.3 page 419). This is achieved by creating a deployment schedule entry in the `ServerManager` in the `Project` properties (see Chapter 7.5.4 page 380).

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 ServerManager under the menu item “Permissions” in the Project properties (see Chapter 7.4.16 page 339)

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 Chapter 7.5.10.6 page 413 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 personalization. Particularly observe that the group hierarchy in the personalization 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.



11.6.2 Configuration semantics

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:

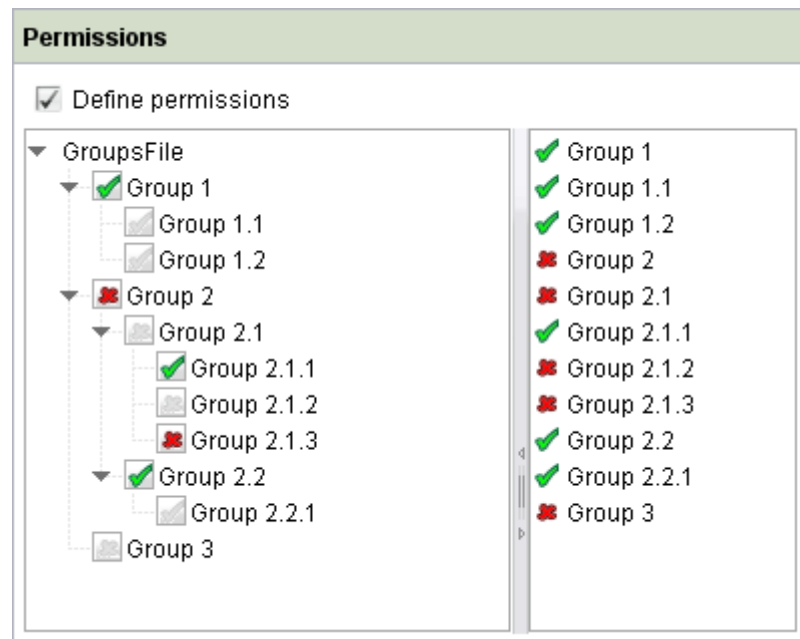


Figure 11-4: Permission component example

The new permission component is capable of assigning permissions for different operations. The operations are basically orthogonal, i.e. there is no scenario “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-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.com/doc/english/properties.html
#*****

# Java command (JDK needed, not JRE)
# absolute path or just "java" when environment variable PATH is set
# correctly
wrapper.java.command=C:\Program Files (x86)\Java\jdk1.7.0_05\bin\java.exe

# Maximum Java Heap Size
# set in MByte with maxmemory=MBYTESNUMBER
# set in percent of total physical RAM with
# maxmemory.percent=PERCENTNUMBER
wrapper.java.maxmemory=700

# Initial Java Heap Size.
# set to same value as wrapper.java.maxmemory to prevent unnecessary Full GCs
wrapper.java.initmemory=700

# Java parameters.
# Gaps in parameter enumeration are allowed since FirstSpirit 5 /
# Wrapper 3.3.6.
wrapper.java.additional.1=-Djava.awt.headless=true
wrapper.java.additional.2=-Djava.security.auth.login.config=conf/fs-jaas.conf
wrapper.java.additional.3=-Djava.security.policy=conf/fs-server.policy
wrapper.java.additional.4=-Dfile.encoding=UTF-8
wrapper.java.additional.5=-Xshare:off
wrapper.java.additional.6=-Djava.net.preferIPv4Stack=true
```



```
wrapper.java.additional.7=-Djava.io.tmpdir=work
wrapper.java.additional.8=
wrapper.java.additional.9=
wrapper.java.additional.10=

# parameter is automatically set by 64bit JVM
# explicitly set as needed by Berkeley-DB:
wrapper.java.additional.11=#-XX:+UseCompressedOops

# Java parameters for garbage collection
# set -Xmn to 40% of wrapper.java.maxmemory
wrapper.java.additional.12=-Xmn280M
wrapper.java.additional.13=-XX:PermSize=200M
wrapper.java.additional.14=-XX:MaxPermSize=200M
wrapper.java.additional.15=-XX:InitialCodeCacheSize=128M
wrapper.java.additional.16=-XX:ReservedCodeCacheSize=128M
wrapper.java.additional.17=-XX:SurvivorRatio=1
wrapper.java.additional.18=-XX:SoftRefLRUPolicyMSPerMB=1
wrapper.java.additional.19=-XX:+DisableExplicitGC
wrapper.java.additional.20=-XX:+UseConcMarkSweepGC
wrapper.java.additional.21=-XX:+UseParNewGC
wrapper.java.additional.22=-XX:+CMSParallelRemarkEnabled
wrapper.java.additional.23=-XX:+CMSClassUnloadingEnabled
wrapper.java.additional.24=-XX:+NeverTenure
wrapper.java.additional.25=-XX:-UseLargePages
wrapper.java.additional.26=-Djava.rmi.dgc.leaseValue=3600000
wrapper.java.additional.27=
wrapper.java.additional.28=
wrapper.java.additional.29=
wrapper.java.additional.30=
wrapper.java.additional.31=
wrapper.java.additional.32=
wrapper.java.additional.33=
wrapper.java.additional.34=
wrapper.java.additional.35=

# Garbage Collector Log
# Logfile fs-gc.log is automatically rotated by FirstSpirit.
wrapper.java.additional.36=-verbose:gc
wrapper.java.additional.37=-XX:+PrintGCTimeStamps
wrapper.java.additional.38=-XX:+PrintGCDetails
wrapper.java.additional.39=-XX:+PrintGCDateStamps
wrapper.java.additional.40=-Xloggc:log/fs-gc.log

# Enable JMX-Connector via fs-server.conf, parameter jmx.port.
# For more JMX parameters see FirstSpirit Administration Manual.

# Timeout parameters in seconds for controlling the Java process.
# Before changing read Wrapper documentation as all timeout parameters
# are related to each other.
# ping.timeout disabled to allow enough time for large heapdumps,
# can be set to 300 if automatic restarts in case of JVM deadlock
# is needed.
wrapper.startup.timeout=30
wrapper.shutdown.timeout=180
wrapper.jvm_exit.timeout=30
wrapper.cpu.timeout=20
wrapper.ping.timeout=0
wrapper.timer_slow_threshold=3
wrapper.successful_invocation_time=35

# Unix: umask for creating files
```



```

# 0022 allow read access to everyone
# 0027 prevent access by others
# 0077 prevent access by group members or others
wrapper.umask=0027

# 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=FirstSpiritServer5 Instance 2

# Display name of the service
wrapper.ntservice.displayname=FirstSpirit 5.0 Instance 2

# Description of the service
wrapper.ntservice.description=FirstSpirit Content Management Server

# Service dependencies. Add dependencies as needed starting from 1
wrapper.ntservice.dependency.1=

# Mode in which the service is installed. AUTO_START or DEMAND_START
wrapper.ntservice.starttype=AUTO_START

# Allow the service to interact with the desktop.
wrapper.ntservice.interactive=false

# Use a console window. This is needed as workaround for Threaddumps
# on Windows. The window will only be visible if
# wrapper.ntservice.interactive=true
wrapper.ntservice.console=true

#*****
# Do not change parameters below as FirstSpirit depends on them
#*****

# Directory to launch FirstSpirit-Server relative to wrapper(.exe)
wrapper.working.dir=../

# Application parameters. Add parameters as needed starting from 1
#wrapper.app.parameter.1=

# Java Classpath (includes wrapper.jar)
wrapper.java.classpath.1=shared/classes
wrapper.java.classpath.2=shared/lib/*.jar
wrapper.java.classpath.3=shared/lib/*.zip
wrapper.java.classpath.4=server/classes
wrapper.java.classpath.5=server/lib/*.jar

# Java Library Path (includes libwrapper.so or wrapper.dll)
wrapper.java.library.path.1=shared/lib
wrapper.java.library.path.2=server/lib

# FirstSpirit-Server Main class
wrapper.java.mainclass=de.espirit.firstspirit.server.CMSServer

# Java Service Wrapper is licensed for redistribution by e-Spirit AG
#include ../conf/fs-wrapper-license.conf

# Do not restart Java VM on failure
wrapper.max_failed_invocations=1

# only one instance of this server with name wrapper.ntservice.name is
# allowed
wrapper.single_invocation=TRUE

# write thread dump if server failed to exit
wrapper.request_thread_dump_on_failed_jvm_exit=TRUE

# return codes used by Webmonitor
wrapper.on_exit.default=SHUTDOWN

```



```
# normal restart:
wrapper.on_exit.23=RESTART
# web-update and restart:
wrapper.on_exit.42=RESTART

# reload this file on Wrapper restart
wrapper.restart.reload_configuration=TRUE

# File which will be monitored every 5 seconds for wrapper commands
wrapper.commandfile=conf/fs-control
wrapper.command.poll_interval=5

# Title to use when running in console mode
wrapper.console.title=FirstSpirit

# no continuous enumeration is needed for wrapper.java.additional.<n>
# and other numbered parameters!
wrapper.ignore_sequence_gaps=true

# set -d64 automatically on operating systems where required
wrapper.java.additional.auto_bits=true

# Server update include file
#include ../server/update/fs-update.conf

*****
# end of file
*****
```

12.2 fs-jaas.conf

```
/*
 * JAAS Login Configurations.
 * (for app-to-config mappings see fs-server.conf, JAAS.*)
 */

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

/* JavaClient without Webstart, without SSO */
plain {
// de.espirit.firstspirit.server.authentication.LdapLoginModule optional
section="LDAP";
de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

/* JavaClient without Webstart, with SSO */
sso {
de.espirit.firstspirit.server.authentication.FSTicketLoginModule sufficient;
// de.espirit.firstspirit.server.authentication.LdapLoginModule optional
section="LDAP";
de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

/* FirstSpirit start page without SSO: WebClient, JavaClient with Webstart */
```



```

webplain {
// de.espirit.firstspirit.server.authentication.LdapLoginModule optional
section="LDAP";
de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

/* FirstSpirit start page with SSO: WebClient, JavaClient with Webstart */
websso {
de.espirit.firstspirit.server.authentication.FSTicketLoginModule sufficient;
// de.espirit.firstspirit.server.authentication.KerberosLoginModule optional
useFullPrincipal="false";
// de.espirit.firstspirit.server.authentication.LdapLoginModule optional
section="LDAP";
de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

/* required additional configuration for KerberosLoginModule: */
/*
com.sun.security.jgss.accept {
    com.sun.security.auth.module.Krb5LoginModule required
        principal=HTTP/fs4host.mydomain.net@MYDOMAIN.NET
        keyTab="/opt/firstspirit4/conf/fs4host-HTTP.keytab"
        useKeyTab="true"
        storeKey="true"
        isInitiator="false"
        doNotPrompt="true"
        debug="false";
};
*/

```

12.3 fs-webapp.xml

```

<?xml version="1.0"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN"
"http://www.eclipse.org/jetty/configure.dtd">

<!--
Configuration of FirstSpirit Web-Server.
For parameter details see http://wiki.eclipse.org/Jetty/ and FirstSpirit Administration Manual.
This file is only used when INTERNAL_SERVLET_ENGINE=1 is set in conf/fs-server.conf.
-->

<Configure id="Server" class="org.eclipse.jetty.server.Server">

<!-- Server Thread Pool -->
<!-- ===== -->
<Set name="ThreadPool">
    <New class="org.eclipse.jetty.util.thread.QueuedThreadPool">
        <Set name="minThreads">5</Set>
        <Set name="maxThreads">250</Set>
    </New>
</Set>

```



```

        </New>
    </Set>

    <!-- HTTP-Connector -->
    <!-- ===== -->
    <Call name="addConnector">
        <Arg>
            <New class="org.eclipse.jetty.server.nio.SelectChannelConnector">
                <Set name="port"><SystemProperty
name="HTTP_PORT" /></Set>
                <Set name="maxIdleTime">30000</Set>
                <Set name="Acceptors">1</Set>
                <Set name="statsOn">false</Set>
                <Set
name="lowResourcesConnections">1000</Set>
                <Set
name="lowResourcesMaxIdleTime">500</Set>
            </New>
        </Arg>
    </Call>

    <!-- HTTPS-Connector -->
    <!-- ===== -->
    <!-- if NIO is not available, use org.eclipse.jetty.server.ssl.SslSocketConnector -->
    <!--
    <New id="sslContextFactory" class="org.eclipse.jetty.http.ssl.SslContextFactory">
        <Set name="KeyStore"><SystemProperty name="cmsroot" />/conf/fs-keystore.jks</Set>
        <Set name="KeyStorePassword">changeit</Set>
        <Set name="KeyManagerPassword">changeit</Set>
    </New>

    <Call name="addConnector">
        <Arg>
            <New class="org.eclipse.jetty.server.ssl.SslSelectChannelConnector">
                <Arg><Ref id="sslContextFactory"/></Arg>
                <Set name="Port">8443</Set>
                <Set name="maxIdleTime">30000</Set>
                <Set name="Acceptors">2</Set>
                <Set name="AcceptQueueSize">100</Set>
            </New>
        </Arg>
    </Call>
-->

```



```

    <!-- Request Log -->
<!-- ===== -->

    <!--
    <Ref id="Handlers">
        <Call name="addHandler">
            <Arg>
                <New id="RequestLog"
class="org.eclipse.jetty.server.handler.RequestLogHandler">
                    <Set name="requestLog">
                        <New id="RequestLogImpl"
class="org.eclipse.jetty.server.NCSARequestLog">
                            <Set name="filename"><Property name="jetty.logs"
default="./logs"/>/yyyy_mm_dd.request.log</Set>
                            <Set name="filenameDateFormat">yyyy_MM_dd</Set>
                            <Set name="retainDays">90</Set>
                            <Set name="append">true</Set>
                            <Set name="extended">false</Set>
                            <Set name="logCookies">false</Set>
                            <Set name="LogTimeZone">GMT</Set>
                        </New>
                    </Set>
                </New>
            </Arg>
        </Call>
    </Ref>
-->

<!-- Set handler Collection Structure -->
<!-- ===== -->
<Set name="handler">
    <New id="Handlers" class="org.eclipse.jetty.server.handler.HandlerCollection">
        <Set name="handlers">
            <Array type="org.eclipse.jetty.server.Handler">
                <Item>
                    <New id="Contexts"
class="org.eclipse.jetty.server.handler.ContextHandlerCollection"/>
                </Item>
                <Item>
                    <New id="DefaultHandler"
class="org.eclipse.jetty.server.handler.DefaultHandler"/>
                </Item>
            </Array>
        </Set>
    </New>
</Set>

```



```
</New>

</Set>

<!-- FirstSpirit Web Applications -->
<!-- ===== -->
<New class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext">
    <Arg><Ref id="Contexts" /></Arg>
    <Arg><SystemProperty name="WEBAPP_ROOT_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_ROOT_URL" /></Arg>
</New>
<New class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext">
    <Arg><Ref id="Contexts" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBMON_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBMON_URL" /></Arg>
</New>
<New class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext">
    <Arg><Ref id="Contexts" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBEDIT5_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBEDIT5_URL" /></Arg>
</New>
<New class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext">
    <Arg><Ref id="Contexts" /></Arg>
    <Arg><SystemProperty name="WEBAPP_STAGING_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_STAGING_URL" /></Arg>
</New>
<New class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext">
    <Arg><Ref id="Contexts" /></Arg>
    <Arg><SystemProperty name="WEBAPP_PREVIEW_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_PREVIEW_URL" /></Arg>
</New>
</Configure>
```

