FirstSpirit™
Unlock Your Content

FirstSpirit™ Release Notes
FirstSpirit™ Version 5.0

Version	**1.31**
Status	**RELEASED**
Date	**2013-08-19**
Department	FS-Core
Copyright	2013 e-Spirit AG
File name	RELN50EN_FirstSpirit_Releasenotes

e-Spirit AG
Barcelonaweg 14
44269 Dortmund | Germany

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

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

e-Spirit

Table of contents

1 Introduction 6

2 Highlights in FirstSpirit 5.0 7

2.1 "My FirstSpirit" – Working individually with the FirstSpirit 5.0 JavaClient 7

2.2 The new WebClient 5.0 11

2.3 Multisite Management – Re-use of content made easy 14

2.3.1 Content Transport 15

2.3.2 CorporateContent 20

3 System requirements 22

3.1 Use of third party products with automatic update management 23

3.2 Web browsers (for use of the FirstSpirit WebClient) 24

3.3 Databases 24

3.4 Operating systems 25

3.4.1 FirstSpirit JavaClient and application for Server and Project Configuration 25

3.4.2 FirstSpirit Server 25

3.5 Java environment 26

3.5.1 FirstSpirit JavaClient and application for Server and project configuration 26

3.5.2 FirstSpirit Server 26

3.6 Web and servlet engine / application server 27

3.7 Integrated preview 27

4 For people changing over from older FirstSpirit versions 28

4.1 New installation (recommended) 29
4.2 In-place upgrade (not recommended) 29
4.3 Required server and project adjustments 32
4.4 Downgrade 33

5 New/Modified functions for all user groups 34

5.1 WebClient 5.0 34
5.2 Improved safety of passwords 35
5.3 New compression mode: Snappy 36

6 New/Modified Functions for Editors 37

6.1 New functions in JavaClient 37
6.1.1 Left-hand area: Search, Explore, Organize 37
6.1.2 Search 38
6.1.3 Clipboard 49
6.1.4 Bookmarks 51
6.1.5 Task list and workflows 57
6.1.6 Revised "New" dialog 58
6.1.7 New browser engine 60
6.1.8 Exclusive editing of data records in content sources 60
6.1.9 Multiple selections 61
6.1.10 Drag and drop 63
6.1.11 Enhanced functionalities in input components 67
6.1.12 CMS_INPUT_LINK: New display of the link input 69
6.1.13 New display of data records (FS_DATASET) 70
6.1.14 Optimized management and display of media 72
6.1.15 URL generation management (search engine optimization) 75

6.1.16 Content Transport ("Add to Content Transport feature") 86
6.1.17 Comparative view in the integrated preview 86
6.1.18 Expanded work area view ("TabView") 87
6.2 New/Modified functions in the WebClient 92
6.3 New/Modified functions in the JavaClient and WebClient 96
6.3.1 Checking inputs ("Dynamic forms") 96

7 New/Modified functions for developers 99

7.1 WebClient 5.0 99
7.2 Template adjustment for Content Highlighting and Easy-Edit 100
7.3 Input components 101
7.3.1 Release of the new input components 101
7.3.2 FS_DATASET: New parameter 102
7.3.3 CMS_INPUT_LINK: New parameter 103
7.4 Dynamic forms (tab "Rules") 103
7.5 Search 104
7.5.1 Global search 104
7.5.2 Search syntax 105
7.5.3 Presentation of search hits (tab "Snippets") 107
7.6 Code completion for forms 107
7.6.1 Adding the input component tags 108
7.6.2 Adding tags, parameters and key terms 109
7.7 Optimized management of media (Exif data) 110
7.7.1 Form and output 110
7.7.2 Conditional display 115
7.1 Template syntax 116
7.2 API extensions 116

7.2.1 FirstSpirit Access API 116
7.2.2 FirstSpirit Developer API 117

8 New/modified functions for administrators 121

8.1 New and changed Java VM and wrapper parameters (`fs-wrapper.conf`) 121
8.2 New security mechanisms for the protection of generated project content 122
8.3 Exclusive editing of data records in content sources 123
8.4 WebEdit settings 124
8.5 Influencing URL generation (search engine optimization) 125
8.5.1 General information on the generation process in FirstSpirit 125
8.5.2 Storing and resetting URLs 127
8.5.3 Standard URL generation 129
8.5.4 Individual URL generation (e.g. "Advanced URLs") 135
8.6 Automatic archiving log files of the Java Garbage Collector 145
8.7 Server update 146
8.7.1 Manual updates 147
8.8 Expanded support for Apache Tomcat web servers 150
8.9 Maintenance mode 152
8.9.1 Maintenance mode schedule 152
8.9.2 Managing the maintenance mode via the Server monitoring 157
8.10 Automatic reporting of software errors 159
8.11 New browser engine 160

9 New / changed functions in modules 161

9.1 FirstSpirit Multisite Management 161
9.2 FirstSpirit BasicSearch (formerly "Search") 164

10 Appendix 165

10.1 Changes in the software behavior 165

10.2 Dropped functions in FirstSpirit Version 5.0 171

10.3 Announcements for future versions 173

1 Introduction

The new functions realized in FirstSpirit V5 are introduced in this document. It is assumed that the reader is already familiar with FirstSpirit and that he has technical background knowledge. To understand Chapter 7 to 9 in particular, in-depth knowledge of the respective field is required (template development, administration).

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.

2 Highlights in FirstSpirit 5.0

2.1 "My FirstSpirit" – Working individually with the FirstSpirit 5.0 JavaClient

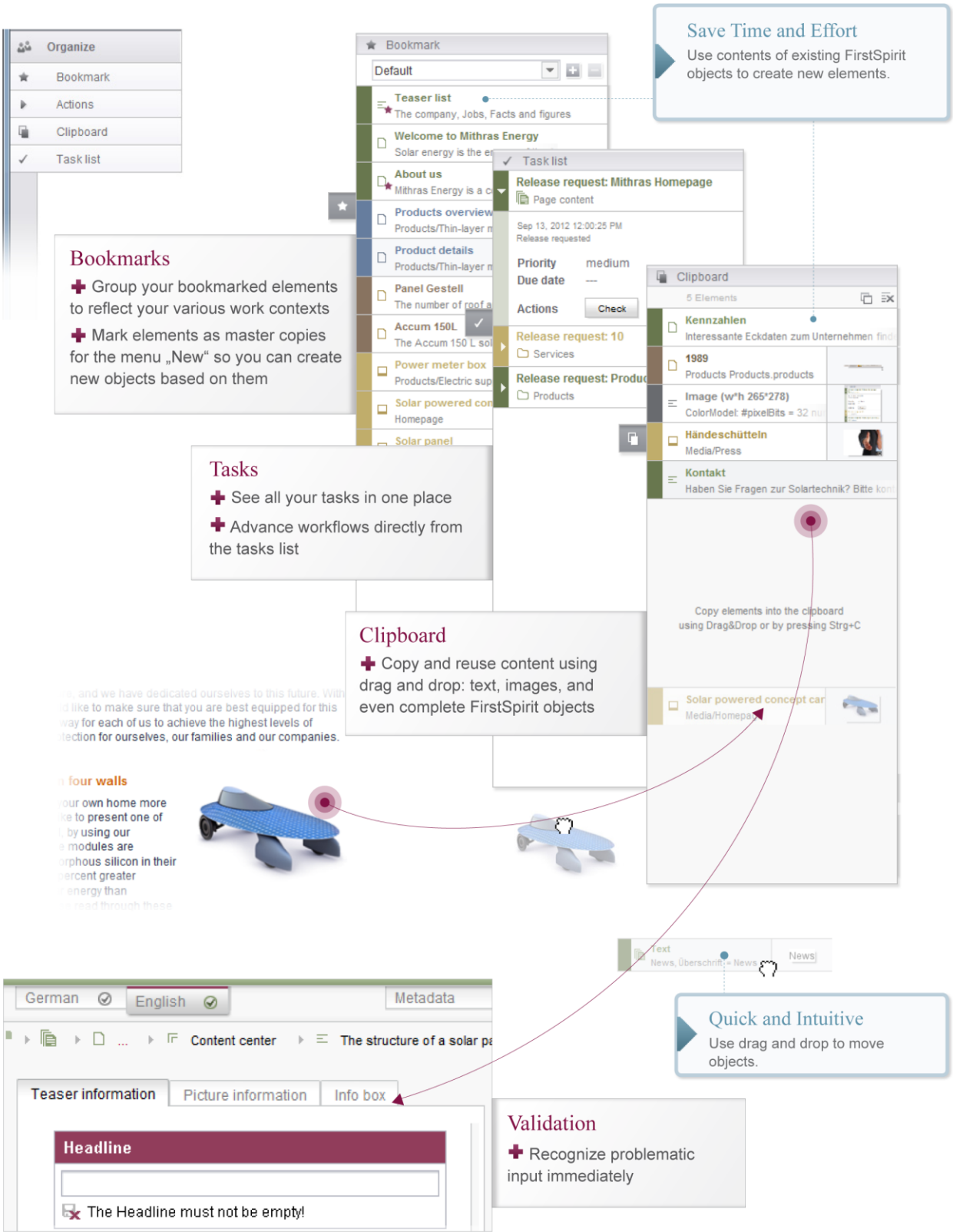
"Organize" – My objects at a glance
You know the classical tree view of FirstSpirit? It is very clearly arranged and efficient if you know at which position of the project an element is.

However, many editors work only in "their" small area of the project. FirstSpirit 5.0 enables you to arrange the most important elements individually in the "Organize" area so that they can be accessed directly.

Bookmarks: My most important working areas in the project only one mouse click away
Every editor works in different subareas of a project. He can arrange his own topics in individual bookmark lists and keep pages, datasets, and media there for a topic at hand for quick access.

Clipboard: Depositing elements briefly and reusing them later on
You would like to note text passages or images down to reuse them later on again? Put elements from FirstSpirit or Microsoft Office comfortably into your personal folder ("Clipboard") by using Drag & Drop.

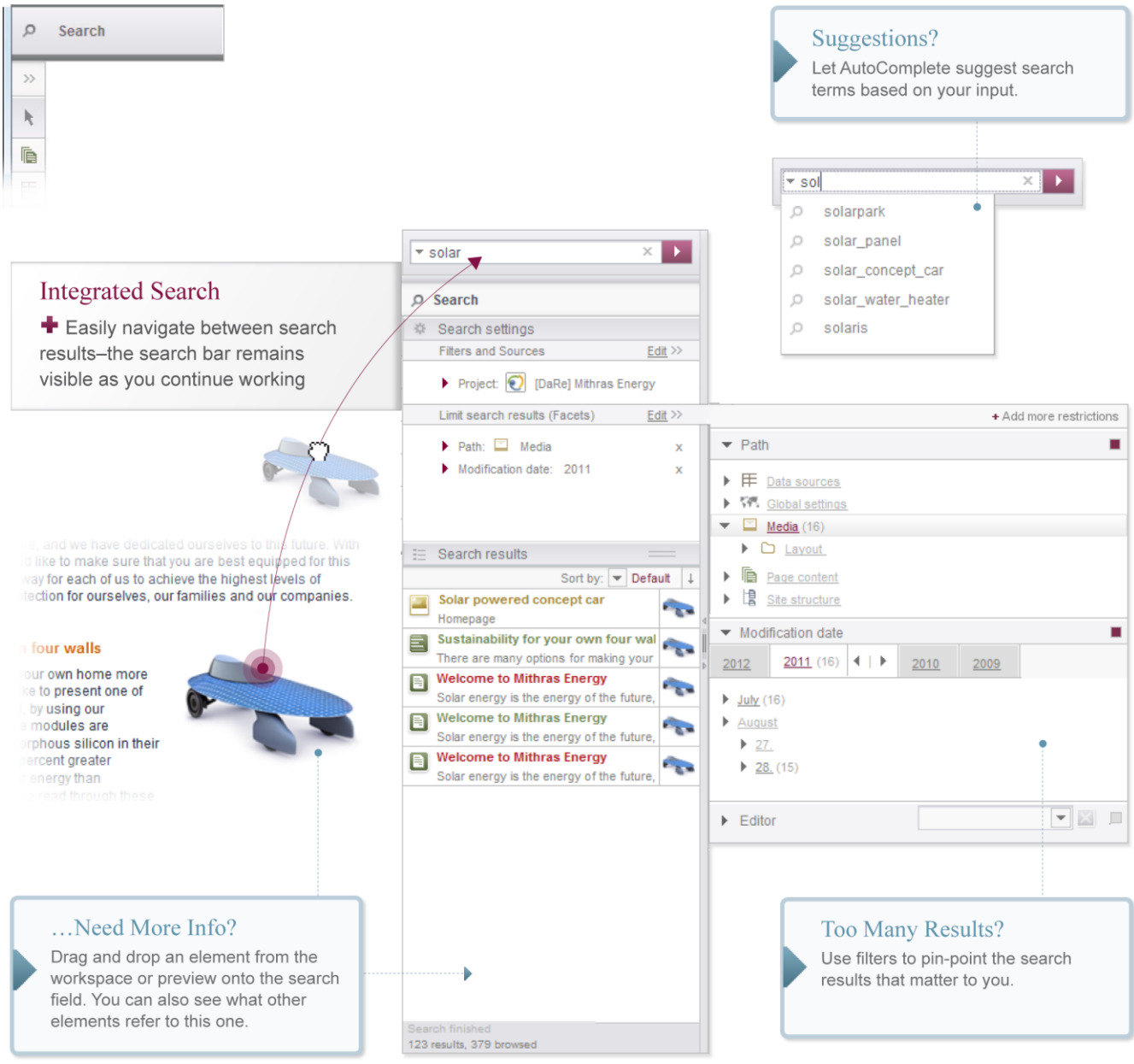
Master copies: My construction kit for frequently used content
Editors mostly have recurring workflows in their project: Entering new staff data into the intranet, creating news, putting products on the website,… It is helpful if not each article or each page must be created totally new. Pre-filled templates make working more efficient: Each editor can compose his individual content kit by means of "master copies" in FirstSpirit 5.0 and reuse modules flexibly.


Organize
Bookmark
Actions
Clipboard
Task list
Bookmark
Default
Teaser list
The company, Jobs, Facts and figures
Welcome to Mithras Energy
About us
Products overview
Product details
Panel Gestell
Accum 150L
Power meter box
Solar powered con
Homepage
Solar panel
Task list
Release request: Mithras Homepage
Page content
Sep 13, 2012 12:00:25 PM
Release requested
Priority medium
Due date ---
Actions Check
Release request: 10
Services
Release request: Produ
Products
Clipboard
5 Elements
Kennzahlen
Interessante Eckdaten zum Unternehmen
1989
Products Products.products
Image (w*h 265*278)
Händeschütteln
Media/Press
Kontakt
Haben Sie Fragen zur Solartechnik? Bitte
Copy elements into the clipboard
using Drag&Drop or by pressing Strg+C
Solar powered concept car
Media/Homepa
four walls
Text
News, Überschrift = News
News
German
English
Metadata
Content center
The structure of a solar pa
Teaser information
Picture information
Info box
Headline
The Headline must not be empty!
Save Time and Effort
Use contents of existing FirstSpirit objects to create new elements.
Bookmarks
Group your bookmarked elements to reflect your various work contexts
Mark elements as master copies for the menu „New“ so you can create new objects based on them
Tasks
See all your tasks in one place
Advance workflows directly from the tasks list
Clipboard
Copy and reuse content using drag and drop: text, images, and even complete FirstSpirit objects
Quick and Intuitive
Use drag and drop to move objects.
Validation
Recognize problematic input immediately

Integrated search with results drilldown

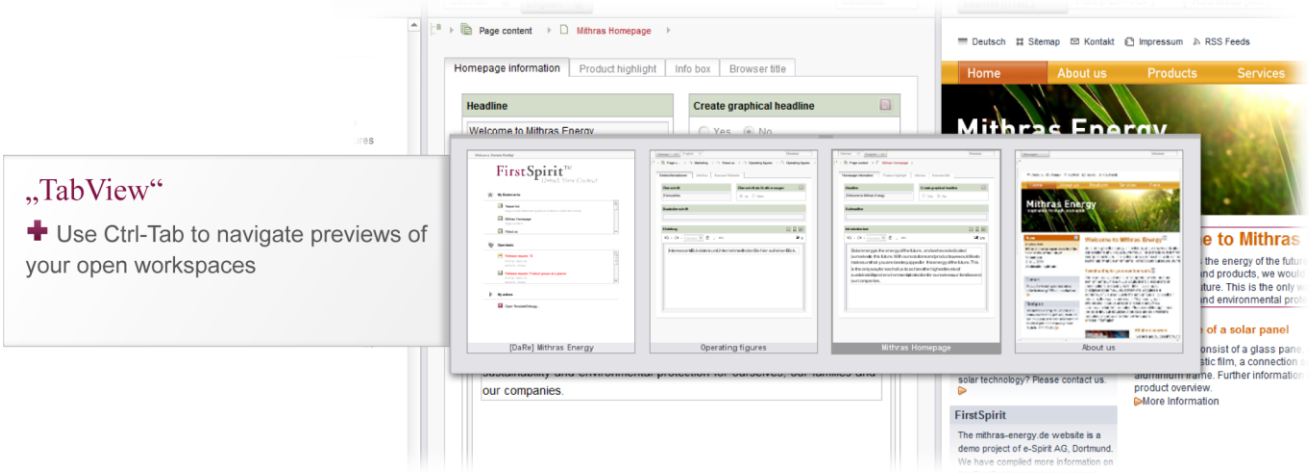
The integrated search provides the quickest way to find pages, articles or media which position in the project you do not know. If the search for a rough keyword delivers too many results, the editor can refine the search criteria subsequently and he can see the reduced results without being obliged to restart the search: Do you want to restrict the search to media, datasets or pages? Do you know perhaps the creation date (year) approximately? With each mouse click the number of hits is reduced immediately and invites you to give it a try.

And even more: Pull an element onto the search field per Drag & Drop to see where in the project it is. And see where it is used furthermore.



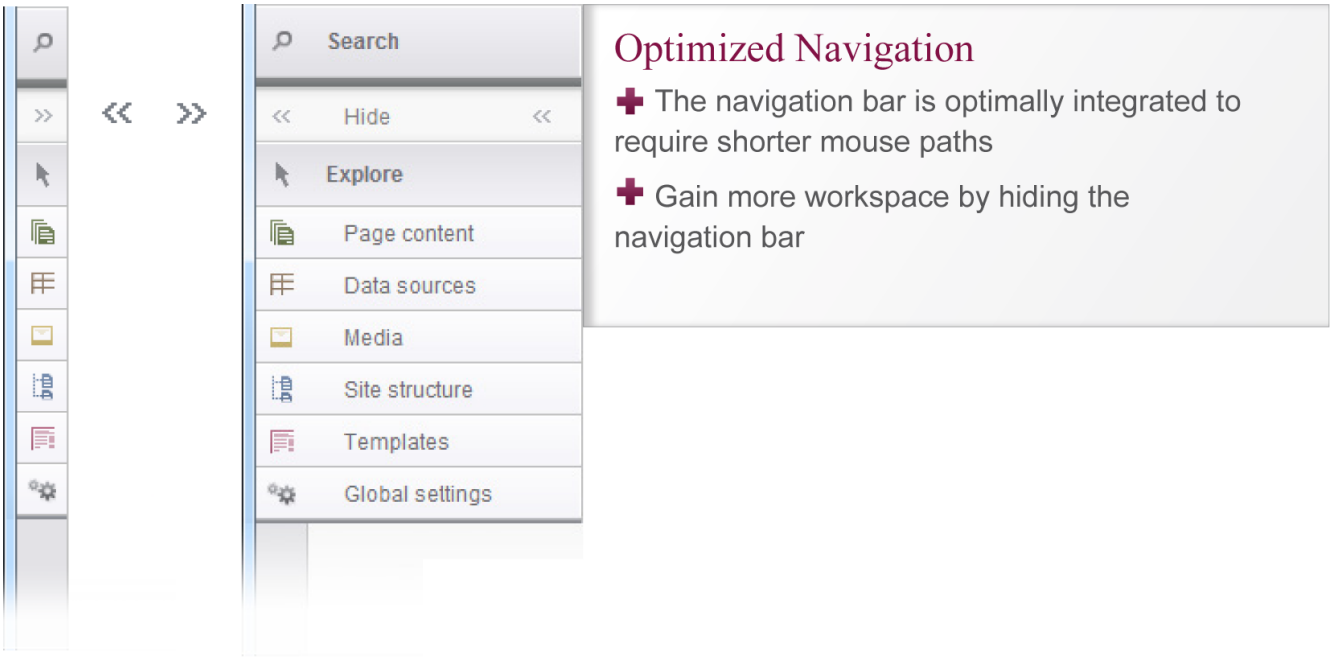
Bird's-Eye View: Overview of my workspaces

JavaClient enables the editor – like modern web browsers – too to use several workspaces on different tabs. The new bird's-eye view of FirstSpirit 5.0 offers you a clearly designed representation of your open workspaces and enables you to switch quickly between these spaces.



An Overview of your work environment

Quick and easy navigation through workspaces



2.2 The new WebClient 5.0

The FirstSpirit WebClient 5.0 was completely redeveloped with a new operating concept and a new code base. True to the maxim "Usability First", the WebClient can be operated immediately – intuitively and without expensive training.

The **user guidance** was subjected to a complete redesign with the following ideas:

- The WebClient can be understood intuitively. Thanks to the design and the functionalities based on quick and simple operation, the use is simply a pleasure. Control mechanisms, which are known also from other modern software programs (e.g. moving of elements per drag & drop), can be transferred directly to the WebClient.

- If there are some questions, they can be clarified quickly and independently by the editor through the orientation tour, the explanatory tool tips and the manual, which are integrated in the WebClient.

Figure 2-1: Tooltips

- Smaller editing changes can be made exactly where they are needed (directly in the text or in the image):

Figure 2-2: Directly working in the browser

- For changes, which require more input by the editor (e.g. creating new pages), the user is guided step by step by form sequences ("wizards").

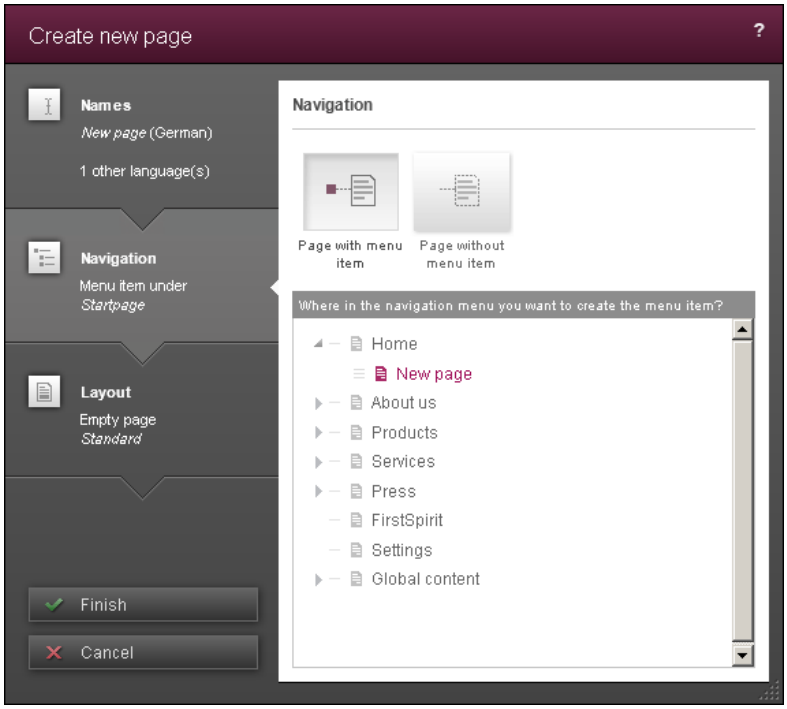


Figure 2-3: Wizard for creating new pages

- The interface is reduced to functions and information, which are actually needed for the daily editing work. This also makes the interface clear and distracts less from the actual website contents. The technical views (e.g. tree structures for different content types and languages), the complex detailed information about the FirstSpirit objects and the special functions are reserved for the JavaClient.

The WebClient is based on innovative **technology**. It offers more security, reliability and compatibility:

- The Google web toolkit (GWT) as basic technology allows high cross-browser compatibility – also in the future – and leads to higher software quality thanks to the good testability.
- The fact that the WebClient runs without flash and Java directly in the browser means high security.
- The WebClient is based on HTML5 and thus becomes a modern web application platform.

The upgrading from earlier FirstSpirit versions is possible with only minimal **migration** costs. Only very few project-based adaptations are necessary in order to be able to use the new WebClient with existing FirstSpirit V4.x projects.

2.3 Multisite Management – Re-use of content made easy

The Multisite Management area contains functions which enable the distribution and re-use of FirstSpirit content in the JavaClient. The user is supported in this task optimally by a comfortable user interface. Content can be reused beyond project and server boundaries.

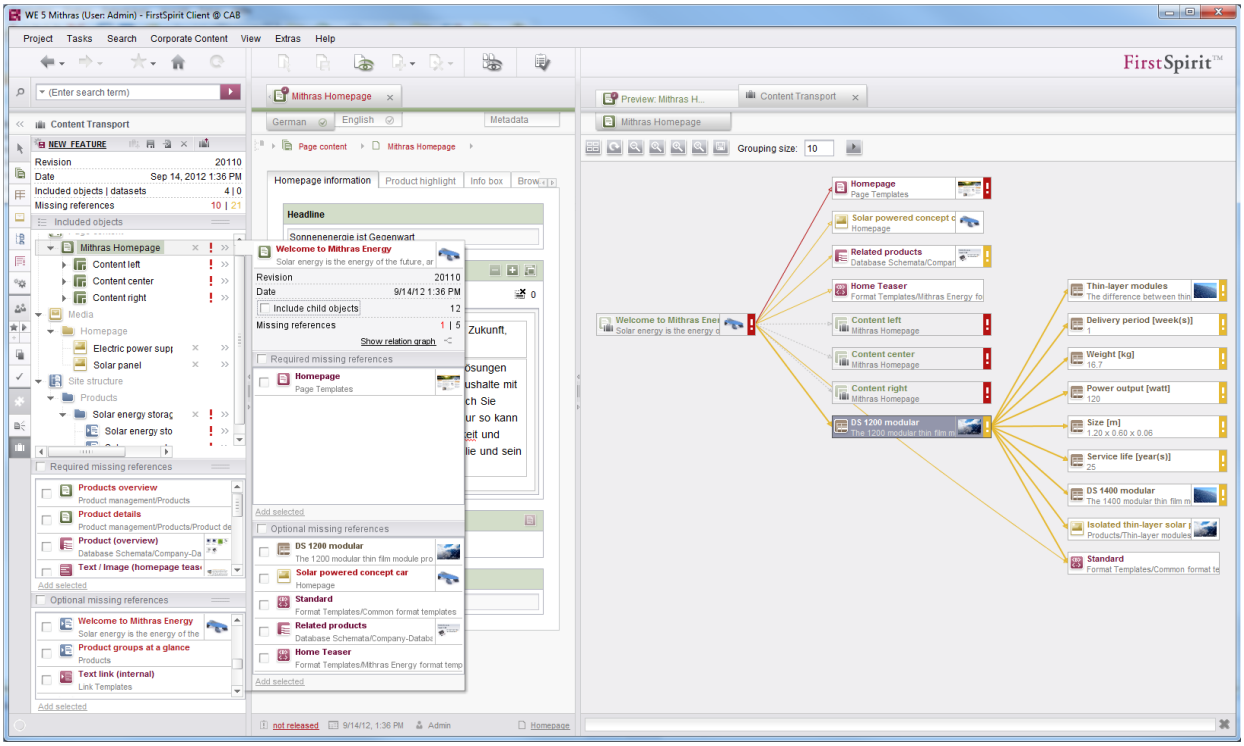
Essential use cases are

- re-use of editorial content and layouts between different projects (sites or tenants)
- simple re-use of particular project solutions
- support for development and quality assurance processes (DQP scenario)

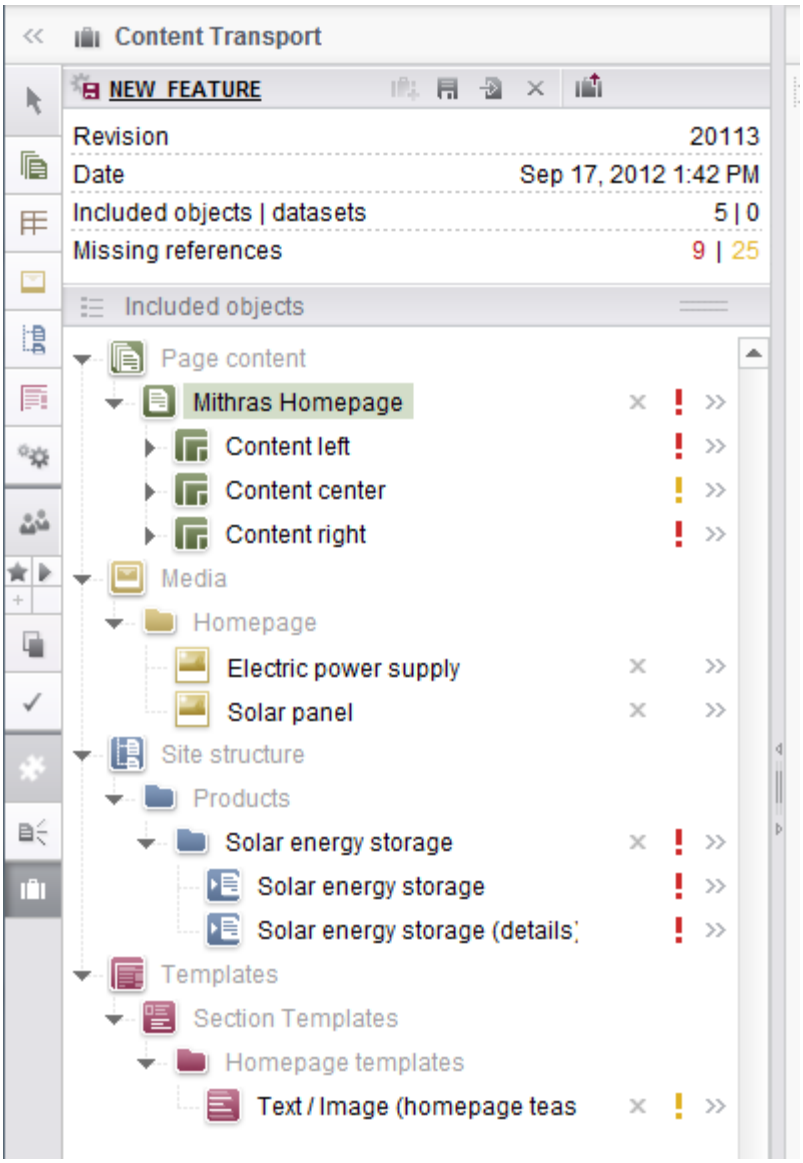
In the new version 5.0, Multisite Management comprises the functionalities *ContentTransport* and *CorporateContent*.

2.3.1 Content Transport

FirstSpirit Content Transport is a complete new development and replaces the function "Template update" which has been known in earlier FirstSpirit versions. By means of the Content Transport, besides the templates all kind of FirstSpirit content can now be transported. All kind? Yes, all, because Content Transport enables you now for the first time to distribute **content deriving from the content store** for re-use.



Which elements does my feature / my compilation contain?

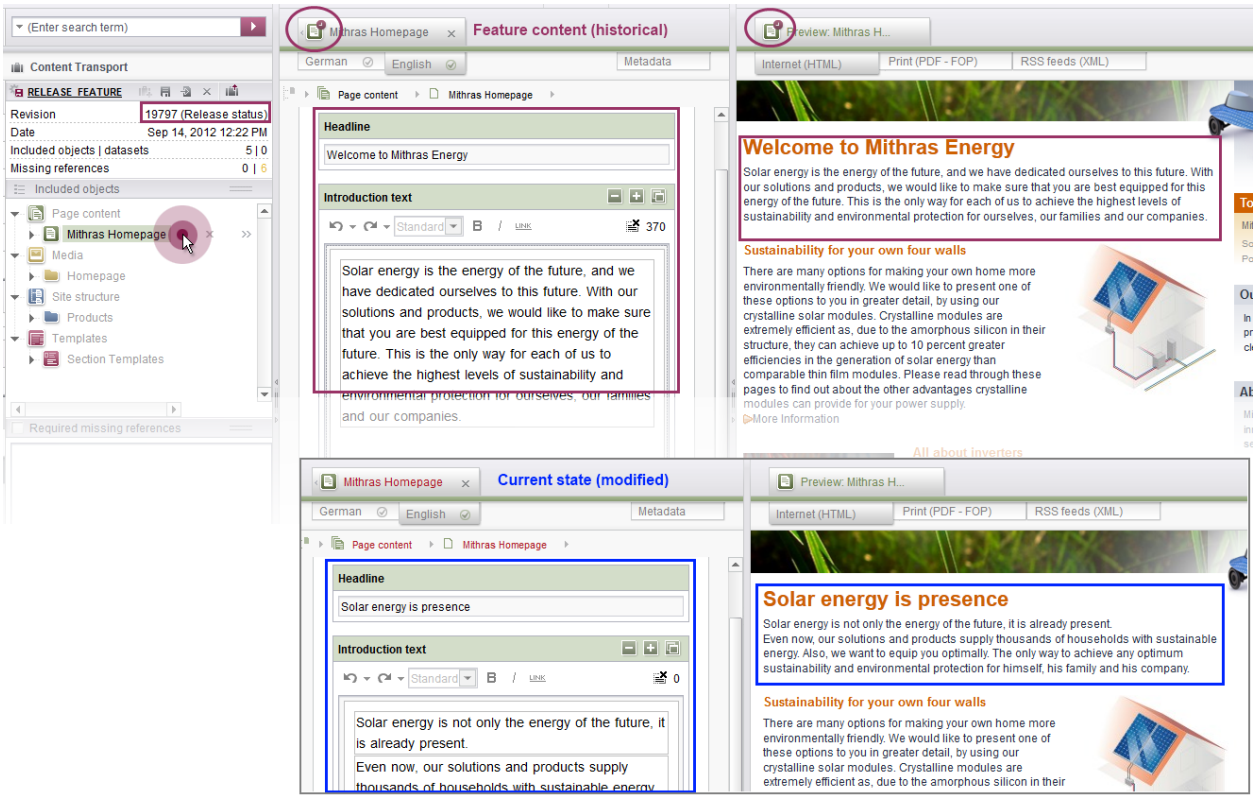


You can see all elements which belong to a feature in the clearly represented tree structure at a glance.

How does the content of the elements of my feature look?

- Historic data in editorial forms and in the integrated preview -

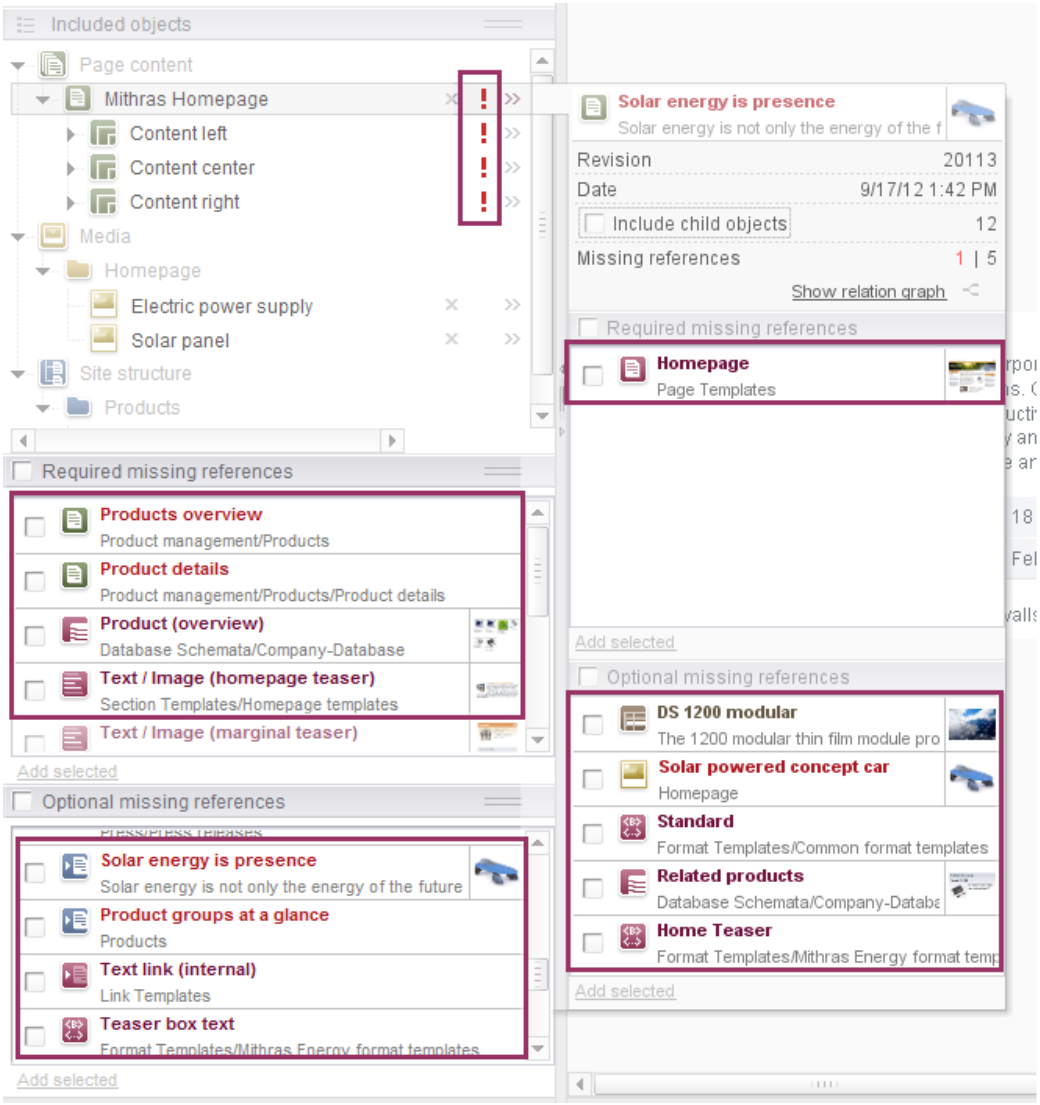
A feature and its elements always refer to a precise project status. This can for example be the release status, but as well a status in the past. FirstSpirit transports only the content of this status. This historic content can be looked at easily at any time. Both, the editorial forms and the integrated preview of the JavaClient show historic data.



Which elements are missing in my feature / my compilation?

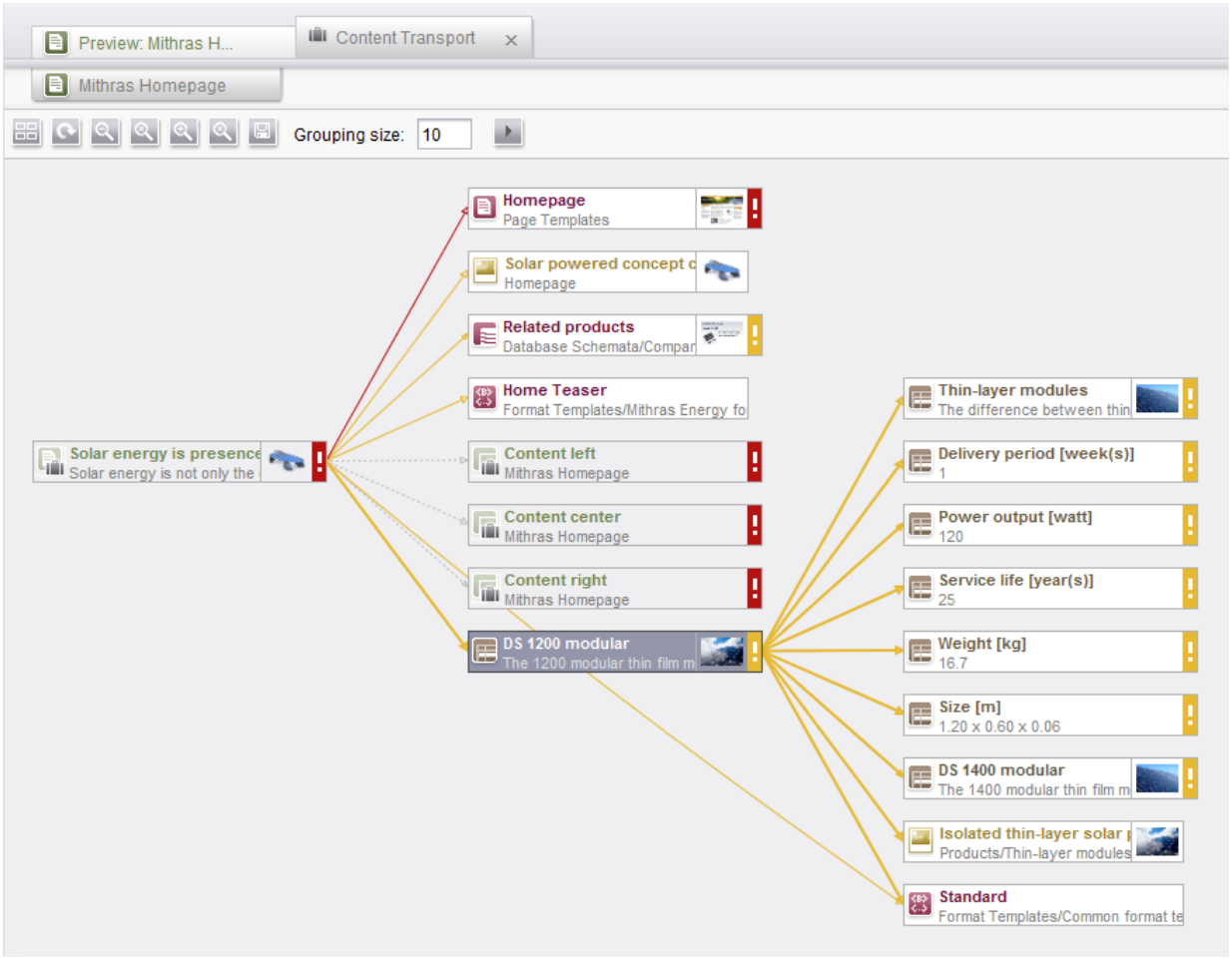
In contrast to CorporateContent packages features need not to be completed, i.e. not all referenced objects need to be contained in a feature because a relationship to existing objects will be established automatically in the target project. The user interface visualizes these open edges at different places:

- directly in **tree view**



The tree view shows missing elements of the feature for the complete feature as well as of single elements which exist already in the feature. The missing elements can be directly added to the feature by the user.

- in the **compilation graph**



The compilation graph offers a general survey of the elements which are existing in the feature and which are still missing (open edges). Assembling packages is optimally supported because elements can be added to the feature or deleted directly by using the graph.

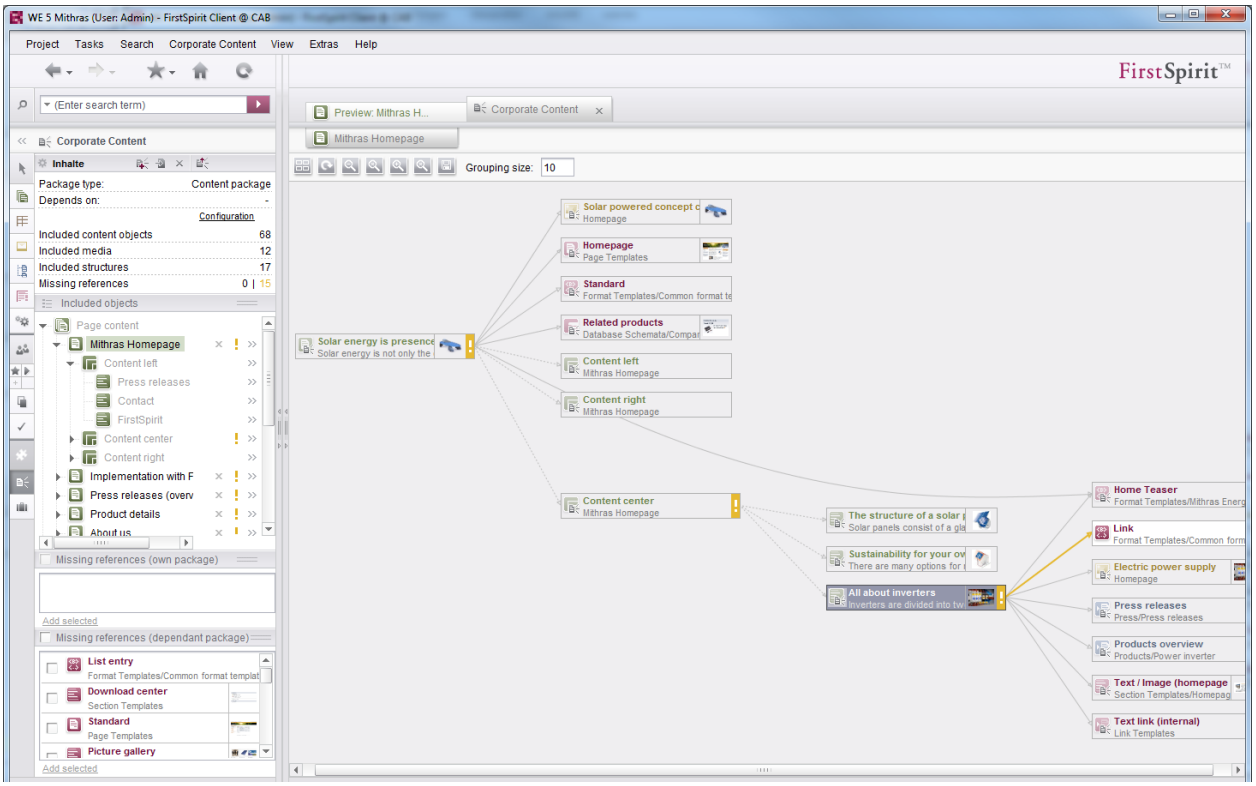
Cross-server transport of features

Compilations from the Content Transport, the so-called "Features", can be transported generally beyond server boundaries (from one server to another). That is why Content Transport optimally supports DQP scenarios and therefore development and quality assurance processes, i.e. processes for transporting new functions from the development system (D) to a quality assurance system (Q) in order to test the new function. After successful tests the function can be transported from D onward to the productive system (P).

The transport of features between DQP systems can be automatized by using the API.

2.3.2 CorporateContent

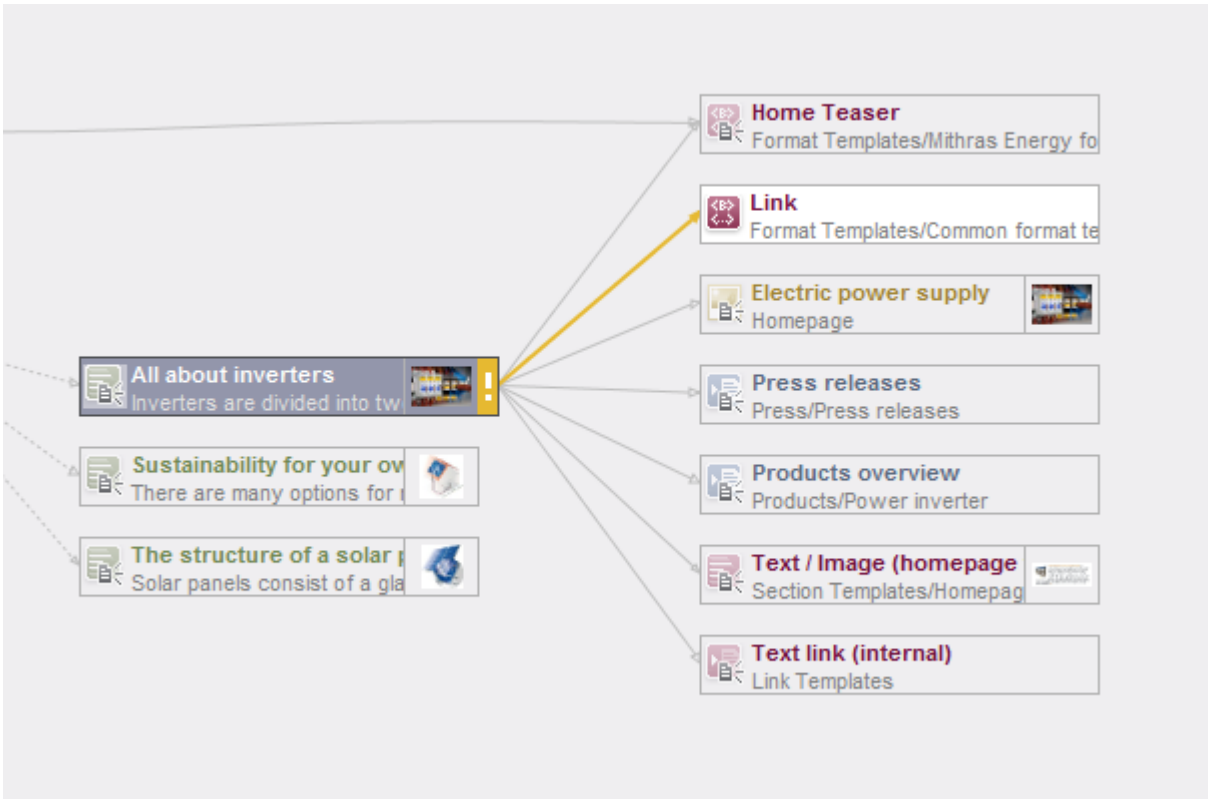
The functionality FirstSpirit CoporateContent was revised extensively in 5.0 and is an advancement of the former "Package management" functionality. With this functionality, templates and content can be distributed automatically between different projects on one server. In this way, content can be re-used with FirstSpirit comfortably across project boundaries. One important aspect when assembling packages is that all dependent objects must be contained in the package too. In FirstSpirit version 5.0, especially the usability for the user (when assembling packages) has been improved in CorporateContent.



The control elements of Content Transport can be retrieved in the user interface of the CorporateContent functionality. Here too the representation of the package content has been simplified by shifting it into the clear tree structure.

Visualizing dependencies to other packages

The compilation graph which exists also in the CorporateContent functionality visualizes dependencies to other packages. If a page is added to a content package for example, the dependency to the templates which are used for this page will be shown. If the respective template does not exist in the belonging template package, this will be marked as a warning.



Creating versions on release states

The possibly most important functional enhancement is the possibility to create packages of the last released state. So, not all elements of a packages must be released no more when creating a version.

3 System requirements

FirstSpirit Version 5.0 (the first version of a major line) is aimed at supporting browsers, data bases and operating systems that are as up-to-date as possible. Owing to the exploding number of new releases, especially in recent times (and especially where browsers are concerned), not all their generations can be provided with the same level of support. The strategy for FirstSpirit Version 5.0 is hence defined as follows:

The implementation is optimized for the current infrastructure, with backward compatibility provided for the respectively previous version, and selectively tested. Previous versions will only be examined if problems occur (in which case debugging is on a goodwill basis only). Given the long development history of FirstSpirit, this provides for very high compatibility in practice, because software products that are legacy today have previously been supported as reference platforms, and any handling processes possibly required are still in the code (i.e. have not been removed).

The reasons for not supporting a great number of legacy systems are hence in no way connected to technical problems, but instead to the effort required for keeping the required test infrastructure permanently available, and for implementing the corresponding tests. This is also why the new category of **Passive Support** is introduced in the *Technical Datasheet* for FirstSpirit 5, which serves to identify systems that are not included in the reference/support list ("**Active Support**"), but are being operated in practice and functioning without known problems. Should any problems with these systems become known, however, these environments will be listed as **Unsupported**.

More information on the involved products and versions that are still being **serviced** or have been taken **out of the service**, respectively, is contained in the following Chapters.

Please see the current Technical Datasheet for detailed information on the system requirements for FirstSpirit Version 5.0 and the categorizations applied.

Some more recent versions of data bases, operating systems, JDKs and HTTP/ application servers will only be released for use with FirstSpirit 5.0 after the initial release of FirstSpirit 5.0. Please see Chapter 10.3 on page 173 for a list of the versions scheduled in this context.

3.1 Use of third party products with automatic update management

The current product maintenance situation of many software manufacturers is that an increasing number of fully-automatic, partly mandatory (i.e. cannot be circumvented by the user or system administration, or only with great difficulty) software updates are performed. Examples: Google Chrome (there the software version is even largely concealed from the user) as well as (to a limited extent) Mozilla Firefox and event Adobe Flash / Reader or Oracle Java. This may appear useful for security aspects, but from the perspective of ensuring interoperability it is problematic, as at any time software update of a third party product can cause incompatibility with FirstSpirit without e-Spirit, as a manufacturer, having any opportunity to react to it in advance.

With FirstSpirit Version 5.0, use of Google Chrome is enabled for the WebClient. Unlike other products (e.g. Microsoft Internet Explorer), Chrome is not only updated fully automatically, but also older Chrome versions are not available at all to download and can therefore also not be used as a "reference version". Therefore, with Version 5.0, e-Spirit will introduce an explicit procedural model for software with forced auto-update (currently only Google Chrome, in future possibly Mozilla Firefox and even the Oracle JDK too): The release tests for the current FirstSpirit version always take place with the current versions of the third party products – therefore, there is NO reference system configuration! If problems occur during the tests, they will either be removed before the release of the FirstSpirit version (which could lead to time delays in the release plan) or the incompatibility will be explicitly pointed out in the release notes. We will then aim to remove the problem for the following FirstSpirit release.

But this procedure does NOT apply to FirstSpirit versions, which are under long-term or medium-term maintenance! The release tests are also performed with the current version of the third party product, but there is no claim to debugging, i.e. if a FirstSpirit version is required, which is compatible with the respective current version of the relevant third party product, then the current FirstSpirit version must also be used. Customers for whom the FirstSpirit long-term or medium-term maintenance is relevant can therefore not use any third party products with automatic update or must implement mechanisms for controlling or circumventing the automatic update management of the third party product.

3.2 Web browsers (for use of the FirstSpirit WebClient)

- **Mozilla Firefox:** See Chapter 3.1 page 23 and *FirstSpirit Technical Datasheet*.
- **Microsoft Internet Explorer:** Version 9 has been included in the maintenance, version 8 continues to be supported. Versions 6 and 7 are not supported.
- **Google Chrome:** With FirstSpirit 5.0 Google Chrome is supported as web browser. For information about supported versions see Chapter 3.1 page 23 and *FirstSpirit Technical Datasheet*.
- **Other browsers**, e.g. Safari, are not supported officially.

For information about the use of Mozilla Firefox and Microsoft Internet Explorer for the Integrated Preview in FirstSpirit JavaClient see FirstSpirit Technical Datasheet 5.

3.3 Databases

- The FirstSpirit internal database, **Apache Derby,** has been updated to the version 10.8.2.2. This Apache Derby database is, however, not suitable for productive use and should therefore be used for tests only.
- **MySQL:** Maintenance for MySQL in version 5.0 has been expired, in exchange, version 5.5 has been included in the maintenance. Version 5.1 is still supported.
- **Microsoft SQL Server:** Maintenance for Microsoft SQL Server 2000 and 2005 has been expired. Version 2008 is still maintained.
- **Oracle:** Maintenance for Oracle databases of version 9 has been expired, in exchange version 11 has been included in the maintenance. Version 10 is still maintained.
- **IBM DB2:** Maintenance for IBM DB2 in the versions 8.2 and 9.4 has been expired. Versions 9.5 and 9.7 are still maintained. IBM DB2 in versions less than 9.4 are not supported.
- **PostgreSQL:** Maintenance for PostgreSQL of the versions 8.0 to 8.3 has been expired, in exchange version 9.1 has been included in the maintenance. Versions 8.4 and 9.x are still maintained.

3.4 Operating systems

3.4.1 FirstSpirit JavaClient and application for Server and Project Configuration

- **Microsoft Windows:** Microsoft Windows 7 is still maintained. However, support for Microsoft Windows Vista and XP expires. Microsoft Windows 2000 and Windows 95 are not supported.
- **Mac OS X:** With FirstSpirit 5.0 Mac OS X in the version 10.7 is supported, however maintenance for 10.6 has been expired. Version 10.5 is not supported.
- **Linux:** With FirstSpirit 5.0 the use of Ubuntu 12.04 LTS with Gnome3 is officially supported.

Note: Only the basic FirstSpirit functions are regularly tested under these two operating systems within the scope of our quality assurance, which continues to focus on Microsoft Windows. Especially under Linux, there are a large number of interfaces (window managers), for which a complete functional test involves a great deal of time and effort. Therefore, more operating-system related problems can occur during regular work with Mac OS X and Linux than under Microsoft Windows, however, as far as technically possible, these will be removed within the scope of the software maintenance. Due to serious differences to Windows-based systems (especially with respect to interface libraries), there are also, e.g. restrictions in drag and drop functions and with regard to the integration of native applications, e.g. in the Integrated preview, so that the AppCenter API is not supported by Mac OS X and Linux. Apart from the Mac OS version 10.6, version 10.5 can also be used, however, this is not officially supported by FirstSpirit.

3.4.2 FirstSpirit Server

- **Microsoft Windows:** Microsoft Windows Server 2008 and Microsoft Windows Server 2008 R2 are still maintained, however support for Microsoft Windows Server 2003 and Microsoft Windows Server 2003 R2 expires.
- **Debian GNU/Linux:** Debian GNU/Linux in version 6 has been included in the maintenance. Version 5 is still maintained, however maintenance for version 4 expires.
- **Red Hat:** Red Hat Enterprise Linux in version 6 has been included in the maintenance, version 5 is still maintained.
- **Suse Linux:** Suse Linux Enterprise Server in versions 11 and 10 is only supported passively.
- **Solaris:** Solaris 11 has been included in the maintenance, Solaris 10 is still maintained. However, maintenance for version 9 expires.
- **IBM AIX:** IBM AIX in version 7.1 has been included in the maintenance, in version 6.1 IBM AIX is still maintained. However, maintenance for version 5.1 expires.

3.5 Java environment

From FirstSpirit Version 5.0 on Oracle Java 7 (**JDK 7**) is officially supported. Modifications in the configuration of the file `fs-wrapper.conf` have been made in this context. See also Chapter 8 page 121.

 *When switching to JDK 7 all modules which you have created yourself should be created newly against the current FirstSpirit version because of changed signatures, because otherwise error messages of the type* `java.lang.VerifyError` *can occur.*

 *Support for version 5 (**JDK 5**) is dropped completely.*

3.5.1 FirstSpirit JavaClient and application for Server and project configuration

- **Oracle JDK** (each in the 32 and 64 bit version): Version 7 (JDK 7) has been taken newly into maintenance. Version 6 (JDK 6) is still supported, however not beneath Update 19 (1.6.0_19), version 5 (JDK 5) is incompatible with FirstSpirit 5.0.

3.5.2 FirstSpirit Server

- **Oracle JDK** (only in the 64 bit version): Version 7 (JDK 7) has been taken newly into maintenance. Version 6 (JDK 6) is still supported, however not beneath Update 23 (1.6.0_23), version 5 (JDK 5) is incompatible with FirstSpirit 5.0.
- **IBM JDK** (only under AIX, only in the 64 bit version): Version 7 (JDK 7) has been taken newly into maintenance. Version 6 (JDK 6) is still supported, however not beneath SR8, version 5 (JDK 5) is incompatible with FirstSpirit 5.0.

3.6 Web and servlet engine / application server

- **Eclipse Jetty** is delivered as integrated servlet engine/application server with FirstSpirit. In FirstSpirit 5.0 the version of the integrated Jetty has been updated to Version 8. For new installations this is no problem. When upgrading from existing installations a conversion of the configuration files `fs-webapp.xml` and `fs-logging.conf` becomes necessary. This will be carried out automatically. Backup files will be created in this case (e.g. `~fs5\conf\*.bak_DATE`), but special configurations must adapted possibly to the new Jetty format.
- **Apache Tomcat:** Version 7 has been included in the maintenance, version 6 is still supported. However, maintenance for version 5.5 expires.
- **Apache HTTP Server:** Version 2.2 is still maintained.
- **Microsoft Internet Information Server (IIS):** Version 7.5 has been included in the maintenance, version 7 is still supported. Maintenance for version 6 expires.

3.7 Integrated preview

Besides the integration of Microsoft Office and OpenOffice, integration of the free open source software "LibreOffice" is also available from FirstSpirit 5.0.

4 For people changing over from older FirstSpirit versions

FirstSpirit Version 5.0 is a major version. It features comprehensive software extensions in comparison with the last released version that would as a rule call for project changes when upgrading from an earlier version of FirstSpirit. Please see Chapter 4.3 on page 32 for more information on the changes required in each case to ensure that projects can still be run. Information on the additional options and changed functions provided in FirstSpirit Version 5.0 can meanwhile be found in Chapter 7 starting on page 99 (changes in the template development) and Chapter 8 from page 121 (changes in server and project properties).

In addition to this, (especially in the field of the WebClient) some functions have also been dropped in comparison to earlier versions. Please see Chapter 10 on page 165 for more information on this. This chapter is very important, especially for upgrades.

Upgrades from the last released version 4.2R4 to Version 5.0 are the only ones to be supported. Although not principally excluded, upgrades from other FirstSpirit versions cannot be supported.

Information on the conversion measures automatically implemented by the system when upgrading to Version 5.0: FirstSpirit 5.0 includes an update of the **search mechanism** via Lucene and Apache POI. This means, amongst other aspects, that more Microsoft Office files will now be searchable than was the case in earlier FirstSpirit versions, including Office 2007 and 2010, for example. This update makes a complete reindexing of ALL projects mandatory, however, when upgrading to FirstSpirit 5.0. This reindexing is implemented automatically in the background in the case of import (see Chapter 4.1 on page 29) or in-place upgrade (see Chapter 4.2 on page 29). The recalculation of the search index can take some time, depending on the number and size of the projects, and burdens the server. Projects can also be used without index, but in this case the search function will not be available. (Individual projects can be indexed by way of appropriate schedules.)

4.1 New installation (recommended)

To start with, the new FirstSpirit Server Version 5.0 needs to be installed (see also *FirstSpirit Installation Instructions Version 5.0*). Existing projects then need to be transferred from the "old" to the "new" server individually by way of export/import (e.g. project export from a FirstSpirit Server Version 4.2R4 and import to the newly installed FirstSpirit Server Version 5.0) and subsequently adapted to new and/or changed functions in accordance with the release notes in hand.

Downgrades from Version 5.0 to older versions are NOT possible Owing to the extensive changes on the server side, project export/import from Version 5.0 to older versions of FirstSpirit is also NOT possible.

4.2 In-place upgrade (not recommended)

Although in-place upgrades, i.e. upgrades from an earlier FirstSpirit version to Version 5.0 by replacing the JAR file (see *FirstSpirit Installation Instructions Version 5.0*, Chapters 2.6.3 and/or 3.8.2) are possible, they are not officially supported and require manual adjustments of the configuration (see below). In-place upgrades should NOT be performed without the support of e-Spirit. The information contained in this subsection is furthermore only addressed at experienced users of FirstSpirit.

The same also applies in this case, however, i.e. downgrades are NOT supported! *See Chapter 4.4 on page 33 for more information.*

An Inplace-Upgrade can be performed in the following way:
(in this Chapter, the path names are followed by `/;` on the other hand, for Windows systems, `\` must be used in the path names.)

1. A valid FirstSpirit-5 license must be available.
2. Securing the complete FirstSpirit-4 installation, incl. databases
   In particular, the configuration data in the `/conf` directory are necessary for a later adaptation of the updated server.
3. New installation of the FirstSpirit-5 server with the desired installer
   (important for later updates with Package Manager! See *FirstSpirit Installation guide 5.0*.)

4. Stop the FirstSpirit-5 server
5. Stop the FirstSpirit-4 server
6. Renaming the directories:
    - `firstspirit5` **=>** `firstspirit5_TEMPLATE`
      (Note: From the `firstspirit5_TEMPLATE` directory, only several files, which have been changed with FirstSpirit 5.0, are necessary; the directory itself can be later deleted)
    - `firstspirit4` **=>** `firstspirit5`
      (Note: This is the directory that contains the productive FirstSpirit server which is to be updated.)
7. Import the FirstSpirit-5 license
   (in the `/firstspirit5/conf/` directory of the FirstSpirit server to be updated)
8. The directory `/firstspirit5/bin/` must be removed and replaced by the `firstspirit5_TEMPLATE/bin` directory.
9. The following files must be removed from the directory `firstspirit5` and replaced by the corresponding file from the `firstspirit5_TEMPLATE/` directory:
    1. Directory `/conf`:
        - fs-wrapper-license.conf
        - fs-wrapper-license.slave.conf
        - fs-wrapper-license.update.conf
    2. Directory `/conf/install`: (Solaris only)
        - `fs-solaris-smf.xml`
10. All files from the directory `firstspirit5_TEMPLATE/server/lib` must be copied in the `firstspirit5/server/lib` directory.
11. Under Unix, the file owner of "fs4" in "fs5" must be changed:
    `chown -R fs5:fs5 firstspirit5`

The following files must be subsequently adapted:

Important: In order to take over the configuration of the FirstSpirit-4 server, the file from the directory `firstspirit5_TEMPLATE` must always be used as a basis for adaptation of the files specified below. The specific settings of the “old” server (parameters, values) must be added manually in them. The "old" file in the `firstspirit5` directory can then be replaced with the adapted file from the `firstspirit5_TEMPLATE` directory. The “old” file from the `firstspirit5` directory should not be used as a basis because, otherwise, a file with undefined content is generated!

- `/conf/fs-server.conf`:

    - Check the file for references to the old FirstSpirit-4 directories and adapt to the FirstSpirit-5- directories
    - JMX configuration (instead of the fs-wrapper.conf file before). The parameters for that are described in the FirstSpirit manual for administrators, Chapter "fs-server.conf" / "Area: JMX".
- `/conf/fs-wrapper.conf` and, when using the clustering functionality, `conf/fs-wrapper.slave.conf`:
    - Adapt the files with respect to heap size (`initmemory`, `maxmemory`, `Xmn`, `PermGen`)
- `/conf/fs-logging.*.conf`:
    - Eventually the file names must be adapted.
    - If the integrated Jetty is used as application server, the eventual special configurations must be adapted manually to the new Jetty format (see Chapter 3.6 page 27).
- `/conf/fs-jaas.conf`
- `/conf/fs-webapp.xml`

In addition, the following adaptations must be performed:

- If necessary, the configurations of third-party software used with FirstSpirit (e.g. Tomcat) must be adapted.
- Updating of the used FirstSpirit modules
- The project web applications must be redeployed
- Updating of the cluster nodes
- If the demo project "Mithras Energy" is used, the most current version can be secured from the directory `firstspirit5_TEMPLATE/export` and imported to the Server-Start.
- Uninstall the old FirstSpirit-4 package, if FirstSpirit 4 had been installed through a package system (via *.rpm, *.deb or Windows). If FirstSpirit 4 had not been installed through a package system, then remove only the start script "fs4" from the system start environment (`/etc/init.d`).
- Known installations residues: delete the FS4 web applications under `~fs5\web\fs4*` (e.g. `fs4staging`) or eventually remove them from the application server
- The directory `firstspirit5_TEMPLATE` can be deleted (it is not necessary anymore).

4.3 Required server and project adjustments

The following adjustments must be implemented in the FirstSpirit server and/or projects in case of an upgrade

- If the **content highlighting** in the JavaClient and/or **Easy Edit** in the WebClient were used in an earlier version of FirstSpirit and at least one of these functionalities is also to be used in future, the corresponding templates must be adjusted in Version 5.0 before the editorial work on the project in any case. See Chapter 7.2 on page 100 for more information on this.
- A number of input components have been replaced by **new FS input components** in Version 5.0:
    - CMS_INPUT_CONTENTAREALIST
    - CMS_INPUT_CONTENTLIST
    - CMS_INPUT_FILE
    - CMS_INPUT_LINKLIST
    - CMS_INPUT_OBJECTCHOOSER
    - CMS_INPUT_PAGEREF
    - CMS_INPUT_PICTURE
    - CMS_INPUT_SECTIONLIST
    - CMS_INPUT_TABLIST

  As the listed input components will fail in WebClient 5.0 and are deprecated for the JavaClient in Version 5.0, these adjustments should be done in the 4.2R4 server before upgrading to 5.0. See Chapter 7.3.1 on page 101 as well as the FirstSpirit *Release Notes Version 4.2R4*, Chapter 5.1, for more information on converting to the new input components.
- A new format for **link templates** has been in existence ever since FirstSpirit Version 4.2. If any link templates should not have been converted to this format yet, this needs to be done in Version 4.2 using the context menu option "Convert link templates" for the link templates concerned. This function is no longer available in Version 5.0. For more information on converting to the new link template format, see also the *FirstSpirit Release Notes 4.2*, heading "Generic link editors".
- The use of "old" input components, "old" link template types or API calls, for example, can lead to **deprecation warnings.** Log files should be examined after these warnings and the cause(s) eliminated.
- Owing to the large version gap when updating the internal **Eclipse Jetty** application server from version 6 to 8, problems may potentially arise with web components (web applications) that may possibly be configured in the preview and/or staging. In this case the configuration of the Jetty and/or web application must be adjusted.
- It is recommended for all FirstSpirit server updates to newly create all self-produced

modules vis-a-vis the new FirstSpirit version. In addition to this, all the installed modules delivered by e-Spirit and all web applications should also always be updated!

- Rare cases may require a new assignment of **passwords** for system connections and/or individual users who are **not** coming from an external authentication system such as Active Directory or LDAP (see Chapter 5.2 on page 35).
- The support for **Apache FOP** (formatting objects processor) in FirstSpirit, which for example allows FirstSpirit contents to be output in PDF format, is realized by way of a module. The previously supported version FOP 0.20.5 is now deprecated as from FirstSpirit Version 5.0. This means that the "Apache FOP v0_20_5" module will no longer be included in the installation and that there will be no more bug fixes for it. The "Apache FOP" module can be used instead (see also Chapter10.1, entry "FirstSpirit modules – Apache FOP"). This conversion may require adjustment of templates for the PDF output channel.
- Both the **interfaces** of the FirstSpirit JavaClient and the FirstSpirit WebClient have been changed drastically in parts. Although user-friendliness was prioritized in the development, editor trainings may be required.

4.4 Downgrade

Downgrades from Version 5.0 to older versions are NOT possible

5 New/Modified functions for all user groups

5.1 WebClient 5.0

The following documentation is available for the WebClient:

- WebClient documentation for **editors**:
  - online: accessible via the icon  in the WebClient
  - offline (PDF): *FirstSpirit Online Documentation* / "Documentation" / "For editors" / "FirstSpirit WebClient documentation"

- WebClient documentation for **developers / administrators**:
  - System requirements: *FirstSpirit Technical Datasheet 5*
  - Preconditions, functional scope and restrictions compared to JavaClient: *FirstSpirit Online documentation* / "WebClient 5.0", sub chapter "Requirements", "Functional scope" and "Restrictions"
  - Configuration: *FirstSpirit Documentation for Administrators*, Chapter "Configuration of the WebClient", "WebEdit settings"
  - Plugin development: *FirstSpirit Online documentation* / "WebClient 5.0", sub chapter "Plugin development"

- Notes about the **migration** from FirstSpirit Version 4.2R4 to 5.0:
  - In general: Chapter 7.1 page 99
  - EasyEdit: Chapter 7.2 page 100
  - Input components: Chapter 7.3 page 101
  - Dynamic forms: Chapter 7.4 page 103

Please also see the following Chapters for more information:

- Editors: Chapter 6.2, page 92
- Template developers: Chapter 7.1, page 99
- Administrators / template developers: Chapter 8.4, page 124

5.2 Improved safety of passwords

Safety measures have been increased within the scope of the passwords which are used for and also stored in FirstSpirit. They are now stored with a new method (Sha384[1] with Salt[2]).

Passwords which were created in earlier FirstSpirit versions and stored in FirstSpirit are converted into the new format when the user logs in for the first time to the FirstSpirit 5 server.

In the case of external authentication e.g. via LDAP or Active Directory passwords are not stored in FirstSpirit and are not concerned.

When a password is converted a log message containing the name of the concerned user will be output, e.g.

```
INFO 14.08.2012 09:19:31.486
(de.espirit.firstspirit.server.usermanagement.UserDTO): Converted old
password hash for user 'editor'
```

In the case of the following exception concerned passwords must be corrected by the administrator by re-entering in the Server properties:

```
IllegalStateException("Deprecated password hash for user '" + loginName +
"'. Please re-enter passwords.")
IllegalArgumentException("Deprecated password hash argument for user '" +
loginName + "'. Please re-enter passwords.")
```

You can identify all users which still have a password in the old format (SHA1) in the application for the Server and project configuration (e.g. "User" / "Edit" or "Project" / "Properties" / "Users" / "Add"): Users whose passwords are still stored in the old format are displayed with yellow background color in user lists.

Enter the following parameter in the file `fs-server.conf` to prevent users from logging in to the FirstSpirit server with password which are still encrypted with the old method:

```
allowSha1PasswordHashes=false
```

[1] http://en.wikipedia.org/wiki/SHA-2
[2] http://en.wikipedia.org/wiki/Salt_%28cryptography%29

It is recommended to set this value as soon as possible.

5.3 New compression mode: Snappy

FirstSpirit 5.0 provides a new compression algorithm for communications between FirstSpirit clients and servers using Snappy for the following operating systems (32 and 64 bit):

- Microsoft Windows
- Linux
- Mac OS

This compression mode can be selected on the FirstSpirit start page and is the standard default for newly installed FirstSpirit servers. The "Deflate Speed" compression mode is used as a fallback (e.g. with unsupported operating systems).

When updating from earlier FirstSpirit versions the setting will be set to the default. For this reason, the compression setting should be checked in the Webstart settings in the application for Server and project configuration / Server / Properties / Webstart / "Compression" or Start page / Project / "Compression".

Please see http://code.google.com/p/snappy/ for more information on Snappy.

6 New/Modified Functions for Editors

6.1 New functions in JavaClient

6.1.1 Left-hand area: Search, Explore, Organize

The left-hand area of the JavaClient has always contained the project's tree structure, via which it is possible to navigate to the required elements. With Version 5.0, not only the project's tree structure will be displayed here, subdivided by stores, but also other navigation options:

- Search results and filter options (see Chapter 6.1.2 page 38)
- Clipboard  (see Chapter 6.1.2.7 page 48)
- Bookmarks  (see Chapter 6.1.4 page 51)
- Workflows ("Task List")  (see Chapter 6.1.5 page 57)
- Scripts ("Actions") 

- For information on CorporateContent  (see Chapter 9.1 page 161)
- For information on Content Transport  (see Chapter 6.1.16 page 86)
- Project-specific information

These icons are located in the so-called "vertical icon bar". Click the respective icon to obtain information on the respective area. Click the same icon to collapse or expand the left-hand area again.

The navigation functions are arranged together in different groups:

 Search

 Explore (Tree view)


Organize (User-defined view)

6.1.2 Search


Unlike previous FirstSpirit versions, the search, which is started using the Search window in the main view, displays the search results directly in the left-hand area of the JavaClient, in which the project's tree structure is also displayed:

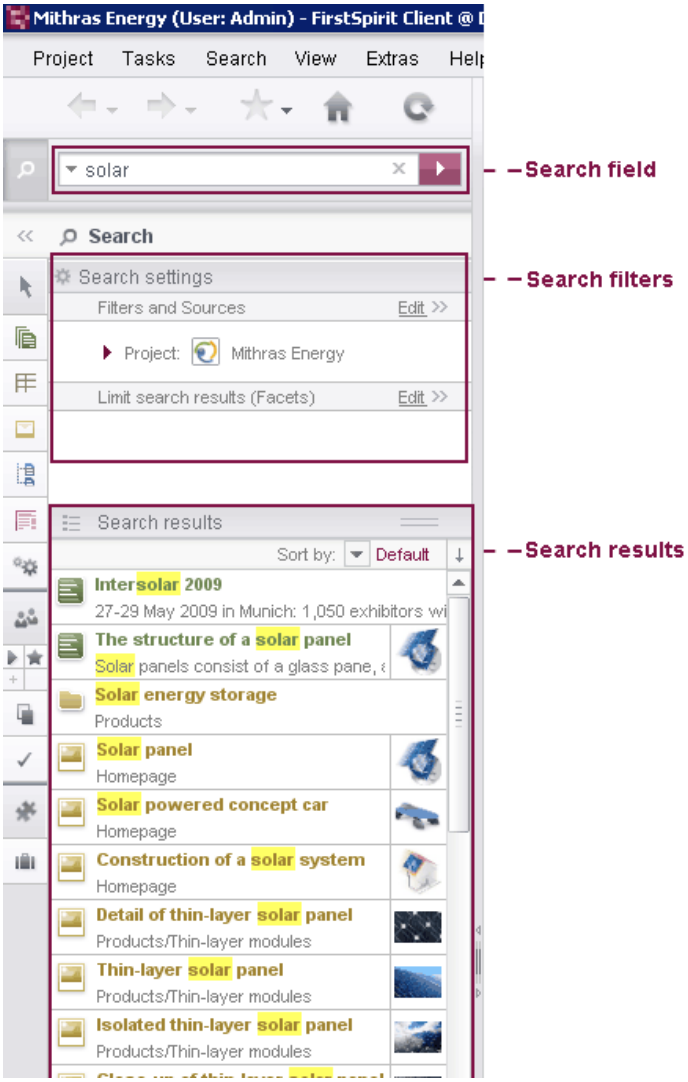


Figure 6-1: New search

6.1.2.1 Search field

The term to be sought is entered in the search field.

The search begins while the first characters are being entered in the input field and shows terms, which begin with the characters entered and are available in the project (auto completion). The number of hits for the suggested term is displayed in brackets. Click a suggestion to copy the term into the search and the corresponding hits are displayed in the results list (see Chapter 6.1.2.3 page 41).

A full text search is performed, which is not case sensitive. Search terms can therefore be entered in upper and lower case letters. If several search terms are entered, the search for these terms is based on an AND operation; only objects that contain all the search terms entered are displayed in the results list. Here it must be noted that, because of the search technology used, search terms with hyphens (-) cannot be used. Instead, the words linked by hyphens should be entered separately, e.g. know how or knowhow instead of know-how.

Not only object names are searched through, but also content (e.g. of pages, data records or media); in the case of media, text in the "Description" field is also searched through.

Not only full-text searches can be carried out, but for example search for input components, which are not filled, search for figures or dates, too. See Chapter 7.5 page 104.

Until now, a full text search did not search the Content Store. Data records can only be searched using a special search in the Content Store itself. With FirstSpirit Version 5.0, the search in the Content Store was completely integrated into the FirstSpirit search, so that it is now also possible to search through the Content Store in the Search / Selection dialogs in which the Content Store can be accessed. Therefore, as in the other Stores, searches are now possible independent of upper/lower case and across several tables.

If you want to search for the term, as it was entered, without using a suggestion, you can start the search with ENTER or by using this icon.

If more than 500 hits are found for a search, a corresponding message appears: "The search returns at least 500 hits. Continue anyway?" If **Yes** is selected, the search is continued. If **No** is selected the search is cancelled; the first 500 hits are then available in the results list. The facets which are described in the following (see Chapter 6.1.2.5 page 44) can be applied only to these hits.

 If the search has been started, it can be stopped by clicking this icon.

If the search is finished, this is indicated in the status bar.

Use this icon to delete the search term from the input field.

With this icon in the search field, search terms for which searches have recently been performed (including those before the current session), can be displayed in a list. Select a search term from the list to perform a search for this term. Use **Delete last searches** to delete the list.

6.1.2.2 Non-text search

Alternatively, FirstSpirit objects can be dragged into the search field with drag-and-drop (for details of drag-and-drop functionalities, see also Chapter 6.1.10 page 63 ff.), for example

- nodes from the tree structure (pages, sections, media, templates, etc.)
- Workspace tab
- Individual input components with content or only their content from an open workspace
- Elements from the integrated preview

Depending on the object, the search then returns, for example, the following:

- Uses of the dropped **Object** in the project, e.g.:
    - Where in the project a medium from the integrated preview is used or maintained?
    - Where in the project is a medium from the tree structure, a workspace or the integrated preview being used?
- Uses of values saved in **input components**, e.g.:
    - Where are right-aligned pictures used?
    - Is a heading or another text being used anywhere else?
    - Which other data records belong to a category?

6.1.2.3 Search results

The search hits are displayed beneath the input field (see Chapter 6.1.2.1 page 39) and the Search Settings (see Chapter 6.1.2.4 page 43 ff.).

Figure 6-2: Search results

They are listed in the left-hand column with the object icon, title and a second text row. By default, the title is the name of the object, the second row is the object's path; however, it can also be text parts of the respective object. Further information can be displayed in a third row by selecting a sorting criterion via **Sort by** (see below). If a search hit is a picture from the Media Store, it is displayed on the right as a preview image (thumbnail). If the mouse cursor is held over the search result the thumbnail is displayed enlarged. Depending on the project configuration, pictures can also be displayed for other object types (pages, sections and data records) (see Chapter 7.5.3 page 107).

The search term is highlighted in color.

By default, the results are displayed according to their relevance. They can be sorted as follows:

Sort by: Use this icon to sort the search result by the following criteria:

- Editor (alphabetically)

- Location (alphabetically, incl. Store name)
- Change date
- Criteria selected under "Limit search results" (see Chapter 6.1.2.5 page 44)

If a criterion has been selected the respective value will be displayed in a third row of the respective object.

The original sorting can be restored by selecting the option "Default".

Descending / Ascending: Use this icon to display the search results in reverse order.

The search results can be limited to certain criteria in the "Search Settings" area (see Chapter 6.1.2.4 page 43 and Chapter 6.1.2.5 page 44).

The search dialog can be left open, without restricting further work in the JavaClient. In this way, for example, it is possible to drag-and-drop the search results into the JavaClient, e.g. the editor can drag a medium directly from the search results into the picture input component of a workspace (see Chapter 6.1.10 page 63). The search results are displayed until a new search is performed. After using the icons to change to a Store in the left-hand area or another area, click the Search icon  to switch back to the search results.

6.1.2.4 Search settings (filters and sources)

The entry **Filters and Sources / Edit >>**

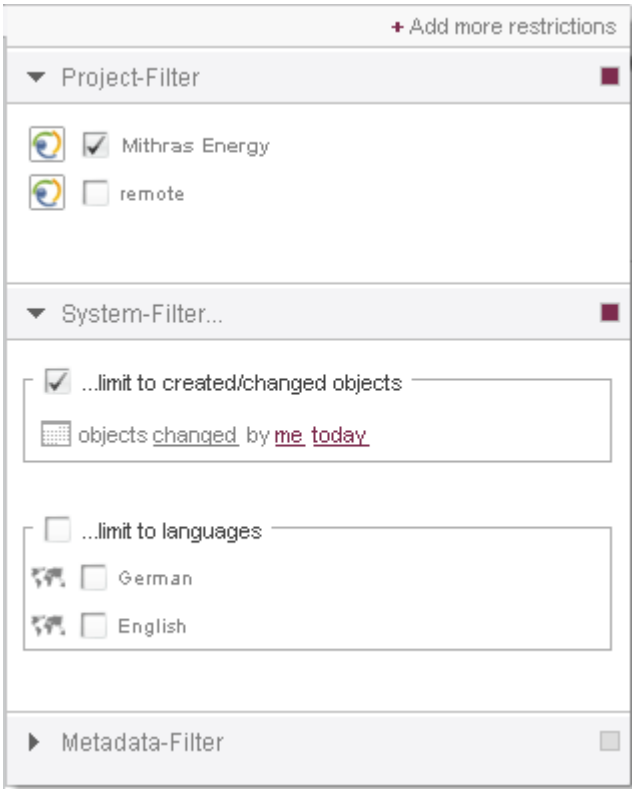


Figure 6-3: Search settings – "Filters and Sources"

can be used to filter the search results, namely by

- **Project filter**
  In this area you can define if the search should be applied to the current project or a remote project.
- **Metadata ("Metadata-Filter")**
  In this area you can search the project's metadata for a specific text. The search functions similar to the "Editor Search" of the "Metadata Search" in the "Search" menu. For further information, see *FirstSpirit Manual for Editors (JavaClient)*, "Metadata Search" chapter.
- **Editor / Change date ("System-Filter...")**
  Special filters can be used in this area: Under **"limit to created/changed objects"** you can select whether the search result is to be filtered by objects, which
  - have been created or changed ("created" or changed")
  - within a specific period ("today", "this week", "last week", etc.)
  - or by a specific user of the project (e.g. "Admin")

- **Languages**
  Search results can be filtered by project language under **"limit to languages"**.

■ Active filters are displayed by a blackberry-colored box next to the filter on the right. The selected filters are also visualized in the "Search Settings" area. Each filter can be disabled by the X next to the filter.

6.1.2.5 Search settings (limiting the number of hits ("Facets"))

With the **Limit search results (Facets) / Edit >>**

Figure 6-4: Search settings – "Facets"

entry, the search results can be filtered, namely by

- **Location ("Search in...")**
  In this area the search results can be limited to a Store area and folders that exist within the Store.

- **Change date**
  In this area the search result can be filtered by change date, i.e. by the date on which the object was last edited. The year is selected first, then the month, day and time. The time can be limited to the nearest hour.
- **Editor**
  The required user can be selected from this dropdown menu. Only search hits edited by the selected user are then considered.

In all displays of search results, the number of search hits is displayed in brackets.

Active filters are displayed by a blackberry-colored box next to the filter on the right. The selected filters are also visualized in the "Search Settings" area. Each filter can be disabled by the X next to the filter.

In addition, **Add more restrictions** can be used to enable the following filters:

- **Created by**
  The required user can be selected from this dropdown menu. Only objects created by the selected user are then considered. (It is possible that this information may not be completely available for projects originating from older versions of FirstSpirit.)
- **Type**
  In this area the search result can be filtered by FirstSpirit object types, e.g. by
  - Content sources
  - Data records
  - Files
  - Media
  - Pages
  - Page references
  - Menu levels ("Site Store folders")
  - Images
  - Sections
  - Section references
  - Page templates
  - Table templates
  - Project settings
- **Released by**
  The required user can be selected from this dropdown menu. Only objects released by the selected user are then considered.
- **File type**
  If a search hit is a medium, the MIME type (*Multipurpose Internet Mail Extensions*) can be

selected here (e.g. "text/plain", "image/jpeg", "application/msword", etc.).

- **Schema**
  If a search hit is a hit from the Content Store, the underlying Database Schema can be selected in this area.
- **Translated / Not translated**
  Use these options to filter for pages and sections of the Page Store for which the option "Page is completely translated to this language" or "include this section in the output" is activated ("translated") or deactivated ("not translated").
- **Creation date**
  In this area the search results can be filtered by creation date, i.e. by the date on which the object was created. The year is selected first, then the month, day and time.
- **Release date**
  In this area the search result can be filtered by release date, i.e. by the date on which the object was last released. The year is selected first, then the month, day and time.
- **File size**
  If a search hit is a medium, the file size can be selected here. The following values are available to choose from
  - EMPTY (no data stored)
  - KB100 (files up to 100 KB in size)
  - MB (files up to 1 MB in size)
  - MB10 (files up to 10 MB in size)
  - MB100 (Files up to 100 MB in size)
- **Release status**
  In this area the search result can be filtered by release status. The following values are available to choose from
  - released
  - not released
  - in workflow
- **Table**
  If a search hit is a hit from the Content Store, the underlying table can be selected in this area.
- **File name extension**
  If a search hit is a medium, the file name extension can be selected here (e.g. png, jpg, pdf, doc, swf, etc.). The file name extension does not necessarily reflect the file type. Therefore, the "File type" filter should be selected for filtering by file type (see above).
- **Meta Data**
  In this area you can select whether only search hits with ("Metadata") or without ("no metadata defined") metadata are to be displayed.
- **Element in Workflow**
  In this area you can select whether only search hits,

  - which are located in a workflow ("Without workflow lock")
  - which are located in a workflow and are locked ("With workflow lock") or
  - which are not in a workflow ("Not in a workflow")
- are to be displayed.

All criteria can also be combined with each other. With each criterion, the result set is further limited. After the required criteria have been selected, the search hits that simultaneously fulfill all criteria are displayed (intersection).

The selected criteria always refer to the currently loaded and displayed search results. If there are more than 500 hits for a search term and if the dialog "The search returns at least 500 hits. Continue anyway?" Was answered by **No** the selected criteria will only be applied to these 500 hits. The result set will not be displayed completely until all hits were loaded (**Yes** in the dialog "The search returns at least 500 hits. Continue anyway?") or filters are used (see Chapter 6.1.2.4 page 43).

The settings can be collapsed or expanded using the icons.

If the filter settings have been changed, which affect an already displayed search result, the filters can be applied by using the "Perform search again" link.

6.1.2.6 Search in selection dialogs

Several input components allow references to be selected from the project, e.g.

- FS_REFERENCE
- FS_LIST
- FS_DATASET
- CMS_INPUT_LINK and
- selection of links in CMS_INPUT_DOM

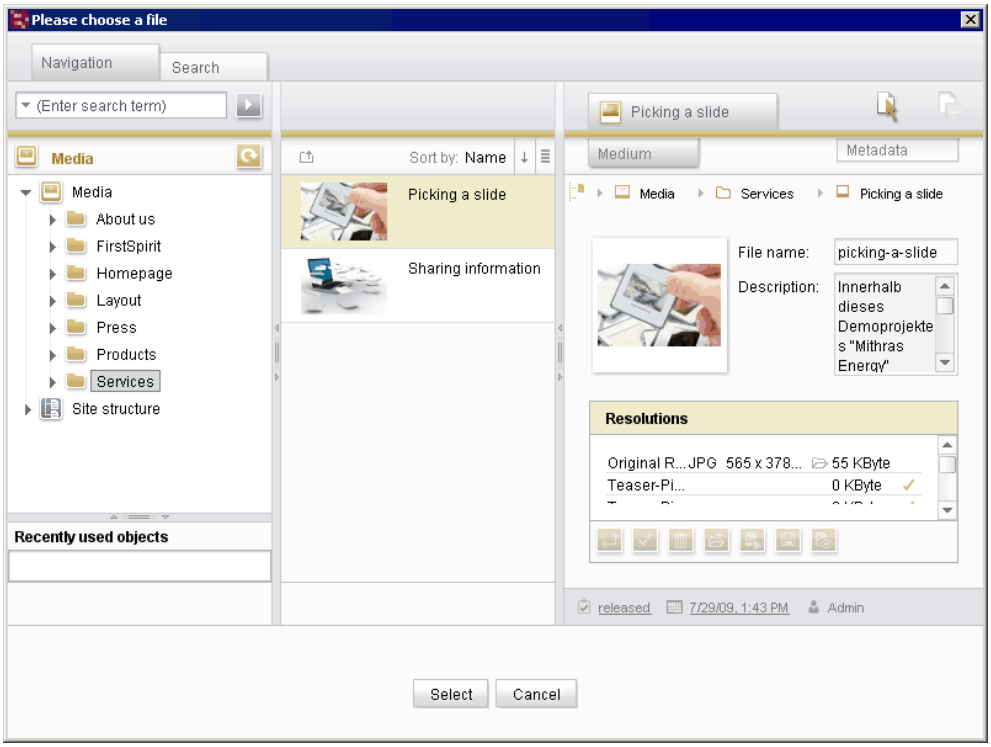


Figure 6-5: Selection dialog

Here too, several new search field functions have been adopted (see Chapter 6.1.2.1 page 39), e.g.

- as soon as the first characters of the search term are entered the first search hits matching the characters entered are displayed,
- the most recently entered search terms can be displayed (icon).

The search results continue to be displayed in the middle column of the "Search" tab.

6.1.2.7 "Search" menu

Some of the search functions in the "Search" menu are made redundant by the new search, but they will be retained initially for compatibility reasons. However, the new search, as described in this Chapter 6.1.2, should be used rather than the search functions of the "Search" menu.

6.1.3 Clipboard

The new clipboard can be used to hold objects, so that they can be used elsewhere later; not only FirstSpirit objects, e.g. pages, page references, pictures, but also data records, sections, individual input components or texts, as well as files from the local desktop computer, e.g. pictures and office files, can be temporarily saved on the clipboard. The editor can use the clipboard as a "collection tank", in which they can clearly and centrally collect materials and content needed for operations to be carried out later.

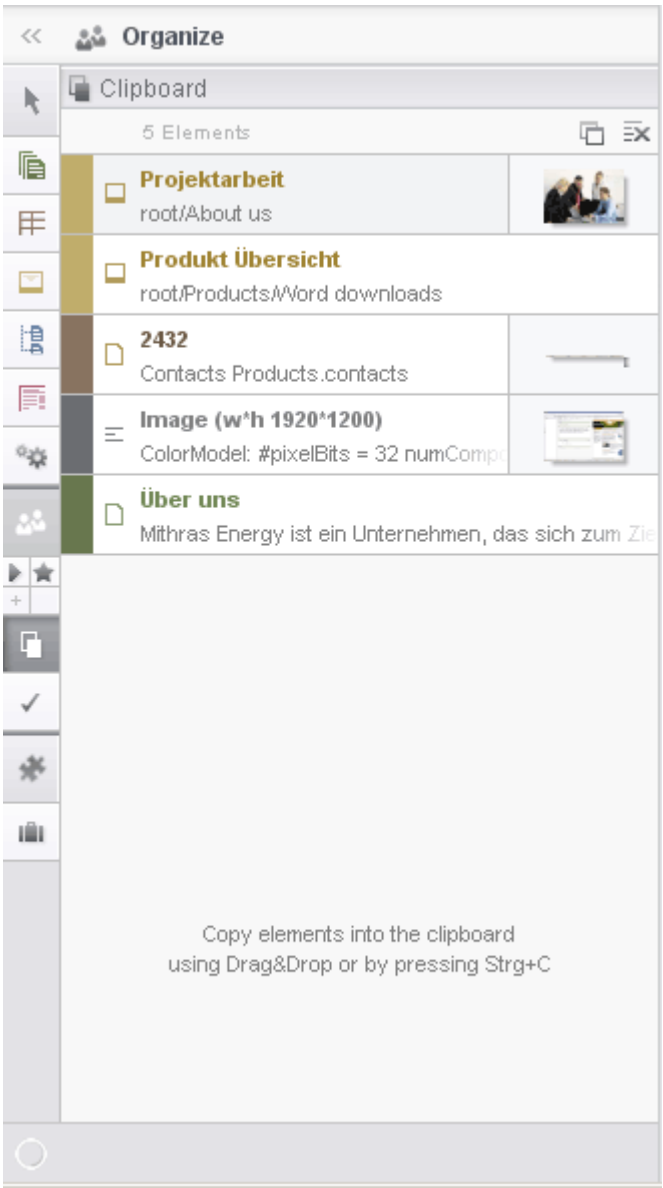


Figure 6-6: Clipboard

The clipboard remains open in the left-hand area until another area is selected using the vertical icon bar or the column is closed using the icon or the separator bar. However, this does not delete the contents of the clipboard and they are still available after changing to another area; the contents remain available until the JavaClient is exited or the clipboard contents are deleted manually.

The objects are listed on the clipboard with a descriptive text, a thumbnail and if applicable an icon and the color of the respective Store. They are included in the list by <Ctrl> + C or the corresponding (context) menu function in FirstSpirit JavaClient or in third party software, or by drag-and-drop. For example, you can drag-and-drop nodes from the tree structure, input components and their content from an open workspace, elements from the integrated preview, etc. into the clipboard. In general, it is also possible to select several objects at once with the <Ctrl> or <Shift> depressed key and then drag them onto the clipboard. The Paste icon indicates that the object(s) can be dropped onto the clipboard. It is also possible to drop them onto the clipboard icon in the vertical icon bar. If the clipboard icon in the vertical icon bar is flashing, this indicates that an object has been copied onto the clipboard, even if the clipboard itself is not open.

Several objects can include several entries on the clipboard, e.g. copied text from an input component can be represented as text and as a picture. The entries can be expanded using the icon on the left-hand side. How many objects are on the clipboard is displayed above the list (e.g. "2 elements").

If the mouse cursor is held over an entry, after a short time a drop-down box opens with an enlarged display of the object.

- (<Ctrl> + <Shift> + V): Use this icon to open the clipboard in a separate window, which remains in the foreground.
- (<Ctrl> + <Alt> + <Shift> + C): Use this icon to delete all the entries on the clipboard.
- : Use this icon to remove the respective entry from the clipboard.
- : If the entry is a FirstSpirit object, use this icon to switch directly to the object. It is opened in a tab in the workspace where it can be further edited.

The content of an entry can also be used elsewhere in the JavaClient by means of drag-and-drop.

6.1.4 Bookmarks

 Bookmarks are most of all familiar from internet browsers and a tried-and-tested means enabling faster user access to important or frequently visited websites.

Bookmarks have also been relied upon in the FirstSpirit JavaClient since FirstSpirit Version 4.1, and have now been further expanded in Version 5.0: their handling has been eased by structuring them in groups. This way the user can create his or her own work environment and is no longer required to change between various stores as often as before.

Most bookmarks can also be tagged as so-called **master copies**. Master copies can be used to quickly create copies of frequently used FirstSpirit objects, including all their subordinate elements and entered contents.

6.1.4.1 Creating bookmarks and master copies

 Bookmarks are created as before by using this icon in the JavaClient tool bar. Master copies are also created using this icon. If the icon is not displayed it can be shown by widening the left column (tree structure) or by way of the  icon at the far left of the tool bar. All object types can be bookmarked on principle. In the case of folders, subordinate objects can subsequently also be invoked in the left-hand column of the JavaClient, but this only applies to the objects located one level below the folder concerned (see Chapter 6.1.4.3 on page 53, section on the vertical tool bar). Master copies can only be created for specific FirstSpirit object types.

Bookmarks and/or master copies can be created for data records by way of the context menu option "New bookmark".

The following dialog will be displayed:

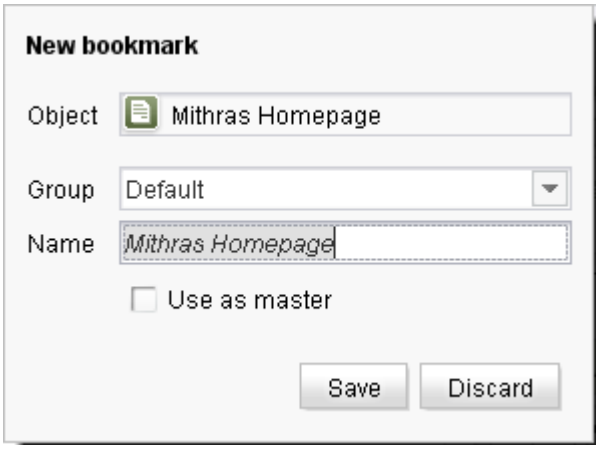


Figure 6-7: Creating a new bookmark

Object: This field displays the selected object the bookmark and/or master copy is to be created for, including the name and object icon.

Group: Bookmarks and master copies can be filed in user-defined groups which need to be created beforehand (see Chapter 6.1.4.2 on page 53). The desired group can be selected from this drop-down list. The group "Default" is provided by default and cannot be deleted. A bookmark or master copy can only be allocated to one group. The allocation to a group can be changed later on as required (see Chapter 6.1.4.4 on page 55).

Name: This field can be used to assign a name to the bookmark or master copy by which it can be located later. It contains the name of the object by default, but this can be changed in this dialog or also later on (see Chapter 6.1.4.3 on page 53, "vertical tool bar" and Chapter 6.1.4.4 on page 55). Names can be used several times over.

Use as master copy: If this checkbox is activated, new objects can be created on the basis of the current object ("Master copy"). Master copies created in this manner can subsequently also be accessed in the JavaClient using the context menu "New" (see also Chapter 6.1.4.5 on page 56). Master copies can also be used in the WebClient (see also the *WebClient Manual for Editors*). Not all object types can be used as master copies and the checkbox may be disabled for this reason.

Save serves to save the bookmark for the selected object along with its name and group allocation. It is then available at the arrow next to the bookmark icon in the tool bar and the corresponding icon in the vertical tool bar.

If the **Discard** button is clicked, no bookmark will be created for the selected object and the dialog will be closed again.

Several bookmarks can be created for one object.

6.1.4.2 Creating and deleting groups

Groups help to easily structure bookmarks in FirstSpirit clients.

Create bookmark group: this icon is used to create new groups. The combobox will turn into a text field:

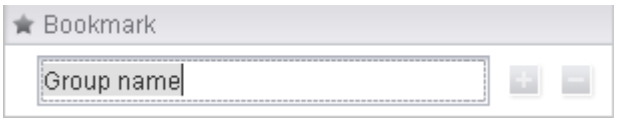


Figure 6-8: Creating a bookmark group

This field can be used to enter a suitable name for the group of bookmarks. The group can then be included in the list of groups by pressing ENTER.

Delete bookmark group: this icon can be used to delete the group currently shown in the drop-down list. All the bookmarks included in the group will also be deleted at the same time. The "Default" group cannot be deleted.

6.1.4.3 Displaying and opening bookmarks and master copies

Bookmarks, master copies and the corresponding objects can be called up and opened in the workspace at the following locations in the JavaClient:

- The arrow next to the bookmark icon in the tool bar:

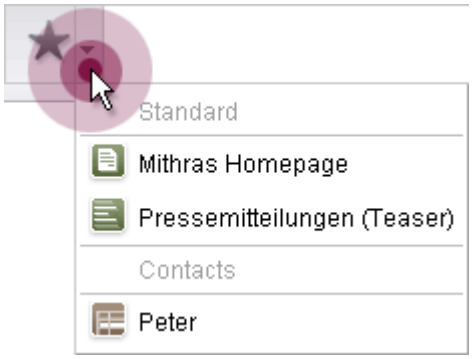


Figure 6-9: Bookmarks / Tool bar

This is where the bookmarks and master copies are listed in groups. If a folder has been bookmarked, only this folder will be shown, but no subordinate folders or objects. Clicking the bookmark will display the object in question in the workspace.

- "My bookmarks" in the project entry page (icon ):

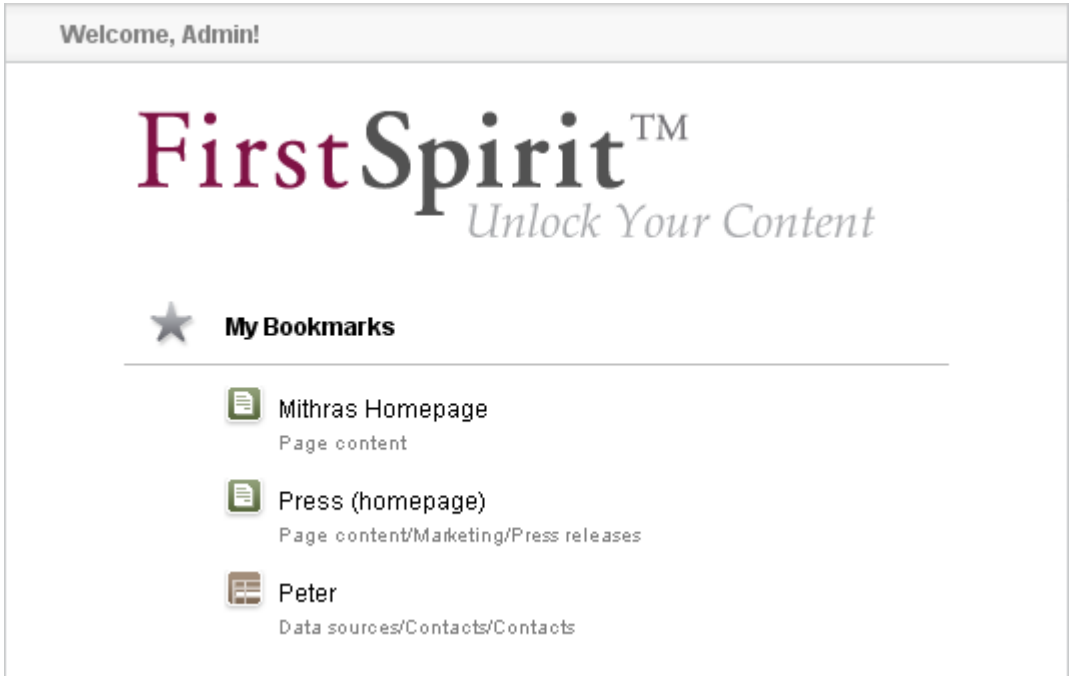


Figure 6-10: Bookmarks / Project entry page

If a folder has been bookmarked, only this folder will be shown, but no subordinate folders or objects. Clicking the bookmark will display the object in question in the workspace.

- The corresponding icon in the vertical tool bar  :

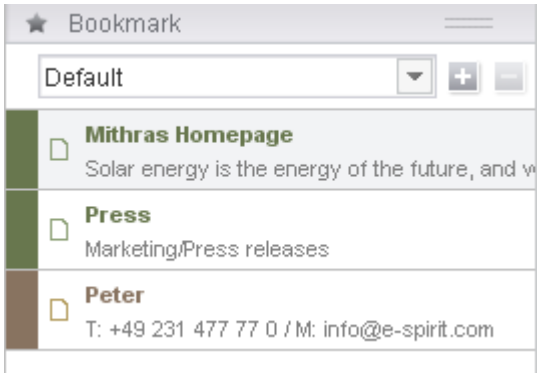


Figure 6-11: Bookmarks / Vertical tool bar

This is where the bookmarks are displayed in groups along with their name, path and/or supplementary text and a preview image ("Snippets", see Chapter 7.5.3 page 107), if applicable. Master copies (see Chapter 6.1.4.1 on page 51, option "Use as master copy") are

identified by an asterisk at the object icon. The respective object can be opened in the workspace with one click on an entry.
To view bookmarks or master copies from other groups the desired group needs to be selected from the drop-down list.
If a folder has been bookmarked, subordinate objects will also be shown here, but only the objects **one** level below the folder concerned. Subordinated elements can also be displayed in the workspace by clicking them.
The name of the bookmarks or master copies can be changed in this view by double-clicking it.
Bookmarks and master copies are listed in their order of creation by default, but this order can be changed by means of drag and drop.
If the mouse cursor moves across a bookmark or master copy, the icons described in Chapter 6.1.4.4 on page 55 will be shown.

6.1.4.4 Editing and deleting bookmarks and master copies

The bookmarks and master copies opened by way of the vertical tool bar (see Figure 6-11) can be edited using the following icons:

Delete: use this icon to delete the bookmark or master copy. This function is only available for objects that are directly tagged as bookmarks or master copies, and hence not for subordinate objects. If a folder has been tagged as a bookmark or master copy, for example, the icon will only remove this tag from the folder itself, but not from its subordinate folders or objects.

» Edit: this icon will open a dialog like the one in Figure 6-7, where the name, group allocation, and the "Use as master copy" option can be changed. The dialog can be closed without making any changes by clicking "Discard".

6.1.4.5 Using master copies

Bookmarks where the option "Use as master copy" is activated (see Chapter 6.1.4.1 on page 51) can be accessed

- Via the "New" option in the context menu of some nodes in the tree structure

Figure 6-12: Function "New" in tree node context menu

- in the dialog "New" / "Insert section" or "New" / "Insert new page" (see Chapter 6.1.6 page 58) or
- at the icon in the horizontal tool bar:

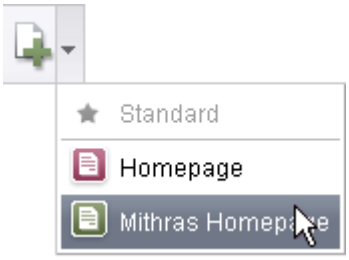


Figure 6-13: Function "New" in horizontal tool bar

Which master copy is being displayed always depends on the store and/or node one is currently located in. If one is located in a page in the **Page Store**, for example, all folders and pages in the Page Store and page templates and pages of the "Global Content Area" that are tagged as master copies will be displayed. In the content areas of pages, only those sections that are allowed for the respective page will be shown as master copies. In a menu level in the **Site Store**, the bookmarked menu levels and page references will be shown.

If a master copy is selected, an exact copy of the object tagged as a master copy will be newly created at the selected position in the tree structure, including all subordinate elements and entered contents. If a Page Store folder has been tagged as a master copy, the "New" function, for example, will create a copy of the folder including all subfolders and subordinate objects. When selecting page templates (from the Template Store), the "New" function will create a new page on the page template which is tagged as master copy.

The name assigned to new objects which are based on master copies is usually the **display name** of the object serving as a master copy. For pages and sections that are based on master copies from the Template Store, however, the **reference name** of the template is used.

6.1.5 Task list and workflows

Upcoming tasks can be displayed also in the Organize area in FirstSpirit 5.0:

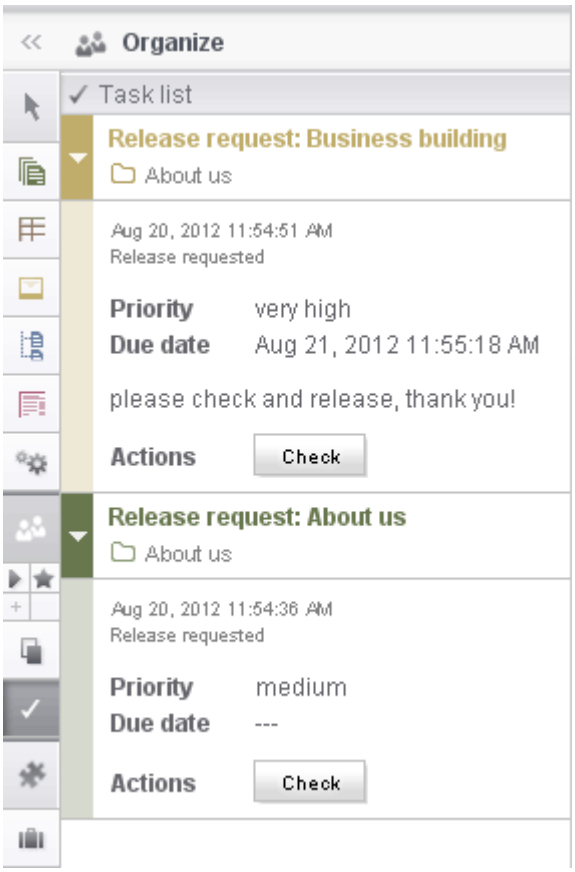


Figure 6-14: Task list

Here, they are listed by date of start or transition with the following information:

- name of the workflow
- name of the element
- color of the object
- path

With double clicking an entry you can open the respective object in the workspace.

When you open the task you will get in addition the following information:

- start time
- status
- priority
- due date
- comment
- action button / next transition

The dialogs for starting and switching workflows to the next state have been optically revised, too.

For more information about tasks in JavaClient also see "FirstSpirit Manual for Editors (JavaClient)", Chapter "Task list" and "Workflows in FirstSpirit JavaClient".

6.1.6 Revised "New" dialog

With FirstSpirit Version 5.0 the dialog for creating new pages and sections in the Page Store has been revised.

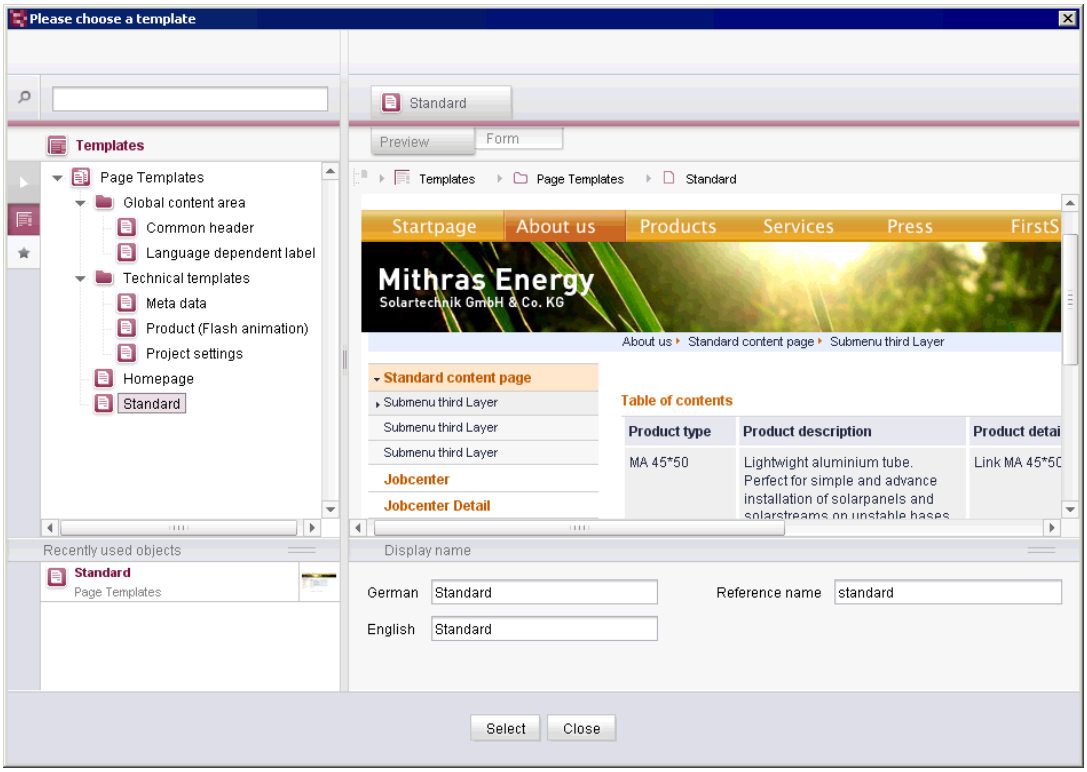


Figure 6-15: Create a new page

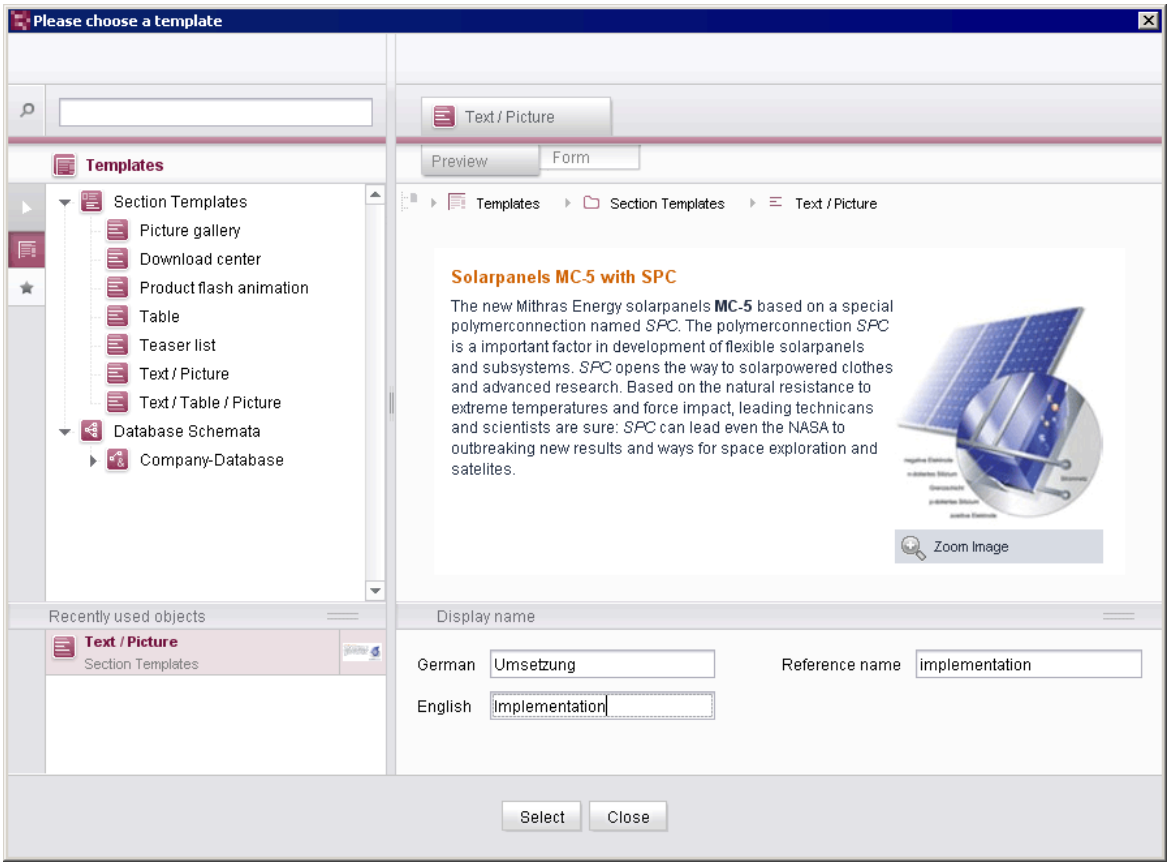


Figure 6-16: Create a new section

Before, for selecting the desired template and assigning a display name two steps were required. These steps are now combined in one dialog (see also *FirstSpirit Manual for editors (JavaClient)*, Chapter "Page Store of the JavaClient" / "Common context menus of the Page Store" / "New"). The dialog has been complemented by the following functionalities:

-  Search by the display name of the desired template
-  Preview of the existing input components
-  Usage of master copies (see Chapter 6.1.4 page 51)

6.1.7 New browser engine

Up to now, the Mozilla Firefox in the Version 3.6 has been used for the integrated preview. With FirstSpirit 5.0 the Mozilla Firefox in Version 15 Beta is available in addition. This can be selected – if admitted by the project administrator – as before via the menu "View" / "Browser engine":

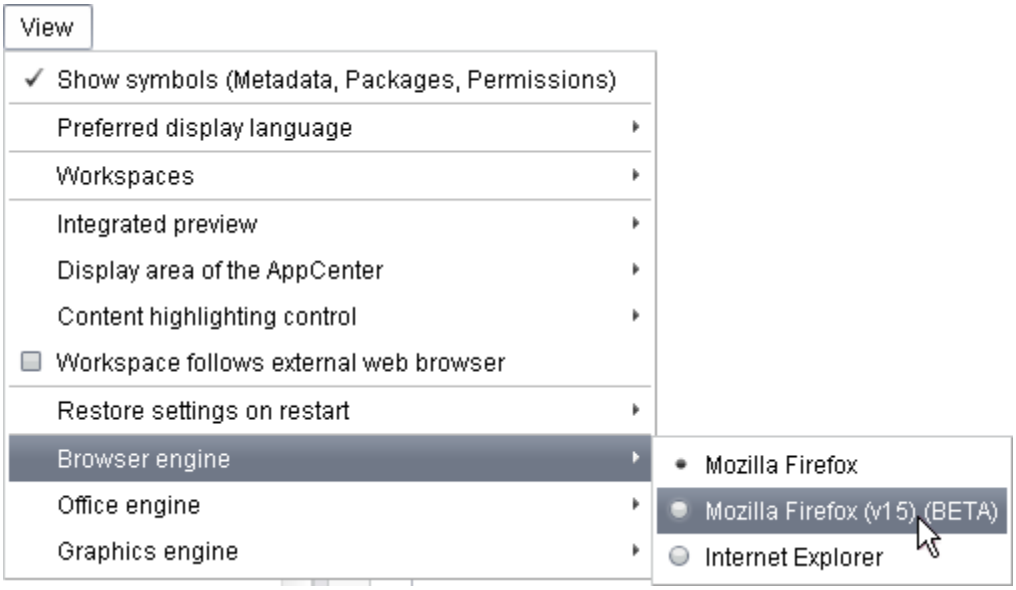


Figure 6-17: New browser engine

This engine is for the time being in the state BETA and is, for this reason, not officially released. Currently, you can switch from one Mozilla engine to the other only with re-starting the JavaClient.

6.1.8 Exclusive editing of data records in content sources

Unlike other Stores, pre-FirstSpirit Version 5.0, content source nodes were not locked when data records were edited. In this way, several editors could simultaneously create new data records, edit, etc. in a content source. Parallel working within a database view is therefore advantageous for multi-user operation, but can result in conflicts, if two editors try to change the same data record at the same time. In this case, the following information is displayed: "The record has been changed by another editor. Your changes could not be saved!" The user can then choose whether their changes are nonetheless to be saved or not (see *FirstSpirit Manual for Editors*

(JavaClient), "Data entry" chapter).

With Version 5.0, a data record can be placed in Edit mode, so that no other editor can make changes to this data record. Edit mode can be activated by

- the  icon of the FirstSpirit tool bar,
- the  icon of the Content Source tool bar
- the "Edit mode on/off" entry in the context menu or
- <Ctrl> + E on a data record

The ID and texts of the data record in Edit mode are always displayed in bold lettering. Only one data record can ever be in edit mode. If the user switches to another data record (data record 2) with the mouse cursor or the  or  icons, the Edit mode for the first data record (data record 1) is deactivated. If changes were made to data record 1, which were not saved, a query appears: "Save changes?" If the new data record (data record 2) is to be edited, it must be placed in Edit mode once again. This means it is not possible to batch process several data records.

If a data record is currently being edited by another editor, corresponding information is provided and the data record cannot be edited.

The option depends on the administrator's settings (see Chapter 8.3 page 123).

6.1.9 Multiple selections

Functionalities concerning multiple selections in the JavaClient have been enhanced in Version 5.0, e.g. to become able to carry out a function on more than one element at once.

Several nodes can now be selected in the tree structure at once easily with the mouse pointer. For this purpose, the left mouse button must be keep pressed:

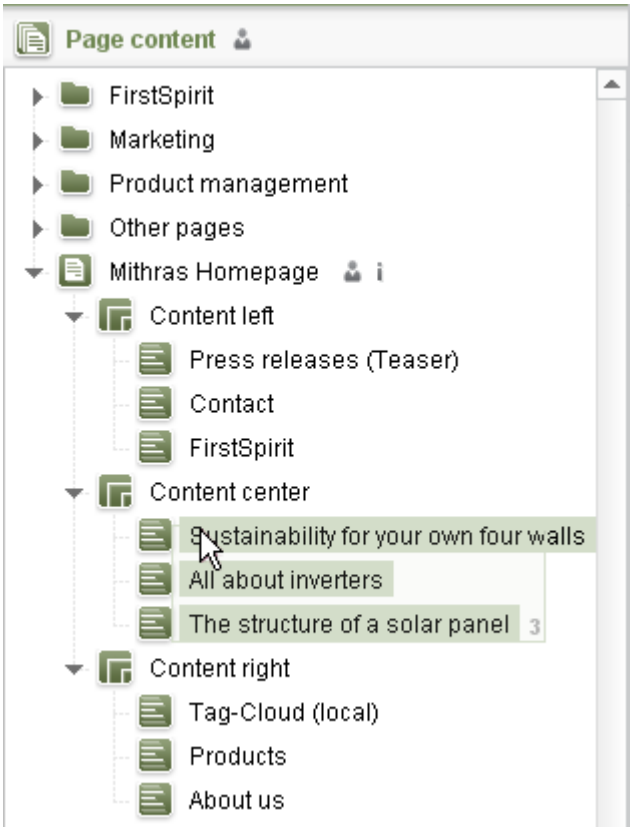


Figure 6-18: Multiple selection in the tree

A frame shows which area is selected, and a figure displays how many elements are selected. Begin the selection from the right hand side best without clicking directly on a node. The selected elements can then for example be – depending on the type of elements – moved or deleted.

This proceeding can be carried out in every store.

Moreover, in the **Media Store**, several elements can be selected at once with the mouse pointer and pressed left mouse button on the tab "Overview", too:

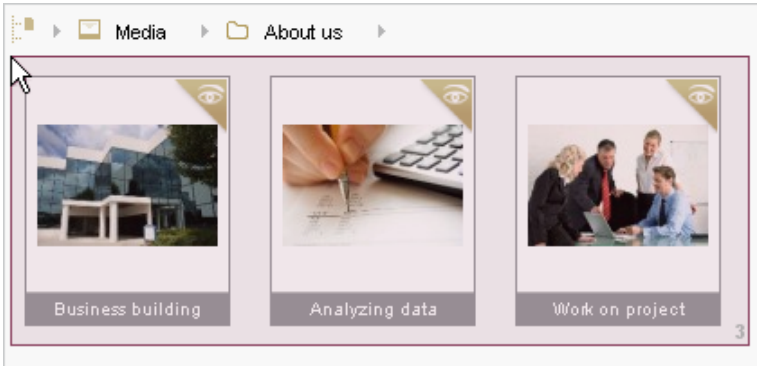


Figure 6-19: Multiple selection in the Media Store

When **Deleting** more than several elements which are marked at once (e.g. by the proceeding which has been described above) not only a confirmation prompt for each single element of the multiple selection will be displayed as before, but with the following dialog all elements of the multiple selection can be deleted at once or each element of the multiple selection can be deleted separately:

Figure 6-20: Deleting a multiple selection

This dialog shows how many elements the multiple selection comprises and the names and ID of the element which is to be deleted currently.

Delete: Only the current element will be deleted.

Skip: The current element will not be deleted.

Delete all: All selected elements are deleted.

Cancel: Nothing will be deleted and the dialog will be closed.

If an element is to be deleted which has got sub-ordinate elements (e.g. in the Page Store: folders with pages, content areas and sections), all sub-ordinate elements will be deleted as well.

"Delete" and "Delete all" can only be carried out if the corresponding element/s are not used in the project ("referenced"). If one of the elements which are to be deleted is still used, a corresponding dialog will be displayed.

6.1.10 Drag and drop

FirstSpirit already supports a large number of options for moving elements into JavaClient or within the JavaClient by means of drag and drop. This intuitive and frequently time-saving option has been developed further in Version 5.0. Whether or not an element may be moved to a specific place by drag and drop is naturally always dependent on the permissions of the respective user and any restrictions that may have been set by the template developer (e.g. for

input components).

Among other things, the following drag-and-drop options have been added in FirstSpirit Version 5.0:

- **Drag and drop search results**
  Depending on the object type, search results from the new search (see Chapter 6.1.2 page 38) can be copied to different places within the JavaClient by means of drag and drop, e.g. into the tree view of the JavaClient, the thumbnail view for media, a workspace or an input component.

- **Drag and drop of input components**

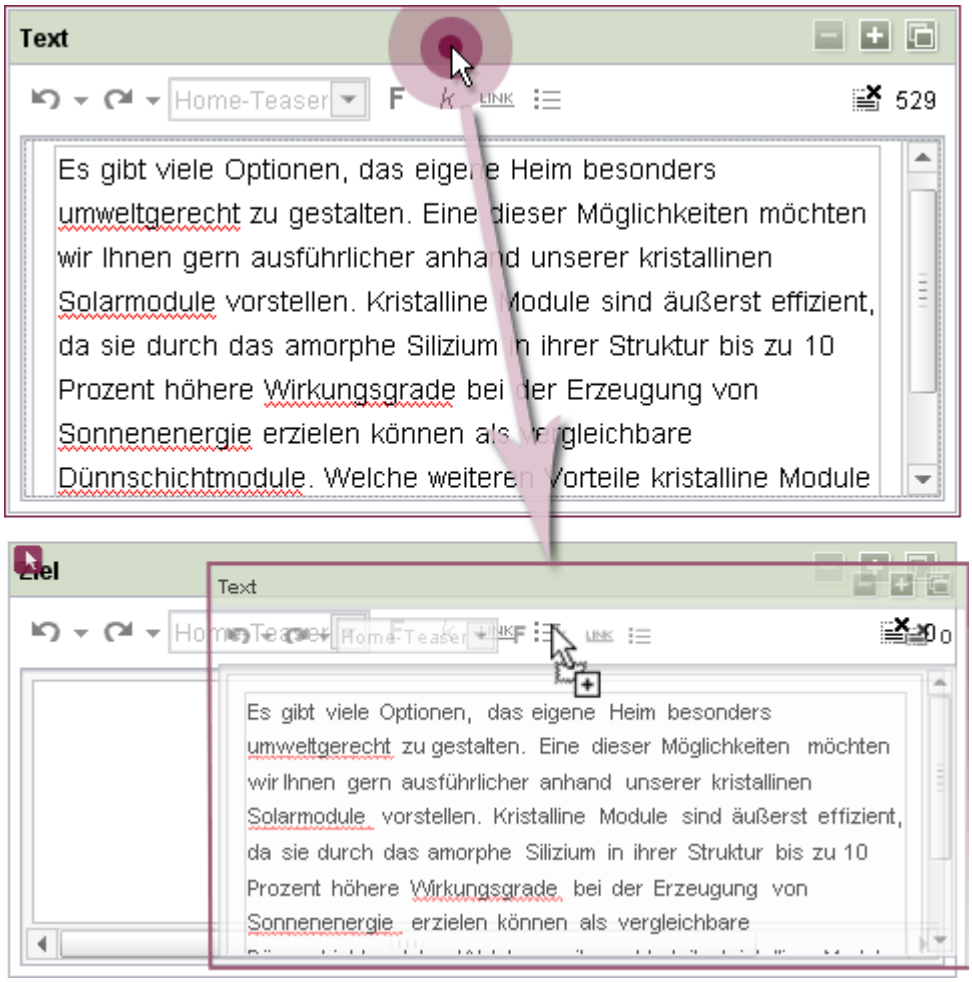


Figure 6-21: Drag and drop of input components

Entire input components can now also be moved by drag and drop. At the same time, the values are also copied from input component to input component. The easiest way to do this is to click the title, with colored background, of an input component and, with the mouse button pressed, drag it to the required target input component. If an input component title does not have a colored background you can, e.g. click the title. A blackberry colored insertion point indicates where the input component can be dropped.

In principle, this is possible at input components of the same type. If the target is an input component, which can accept text (e.g. CMS_INPUT_DOM, CMS_INPUT_TEXT, CMS_INPUT_TEXTAREA), an attempt is made to convert into text and insert the information of the element to be dropped, e.g. the name of a medium, the text of a selected option. If the target is also an input component (drop onto the title), any existing value is overwritten, if the target is e.g. a text field the value is added. In the case of CMS_INPUT_DOM formatting is also copied, provided the configuration allows this.

Other input components, e.g. CMS_INPUT_NUMBER or CMS_INPUT_DATE, can also be

copied and inserted including their content by means of drag and drop.
These drag-and-drop operations can also be made using the

- context menu or keyboard shortcuts (Chapter 6.1.11 page 67) or
- clipboard (Chapter 6.1.2.7 page 48)

- **Drag and drop of content (texts)**

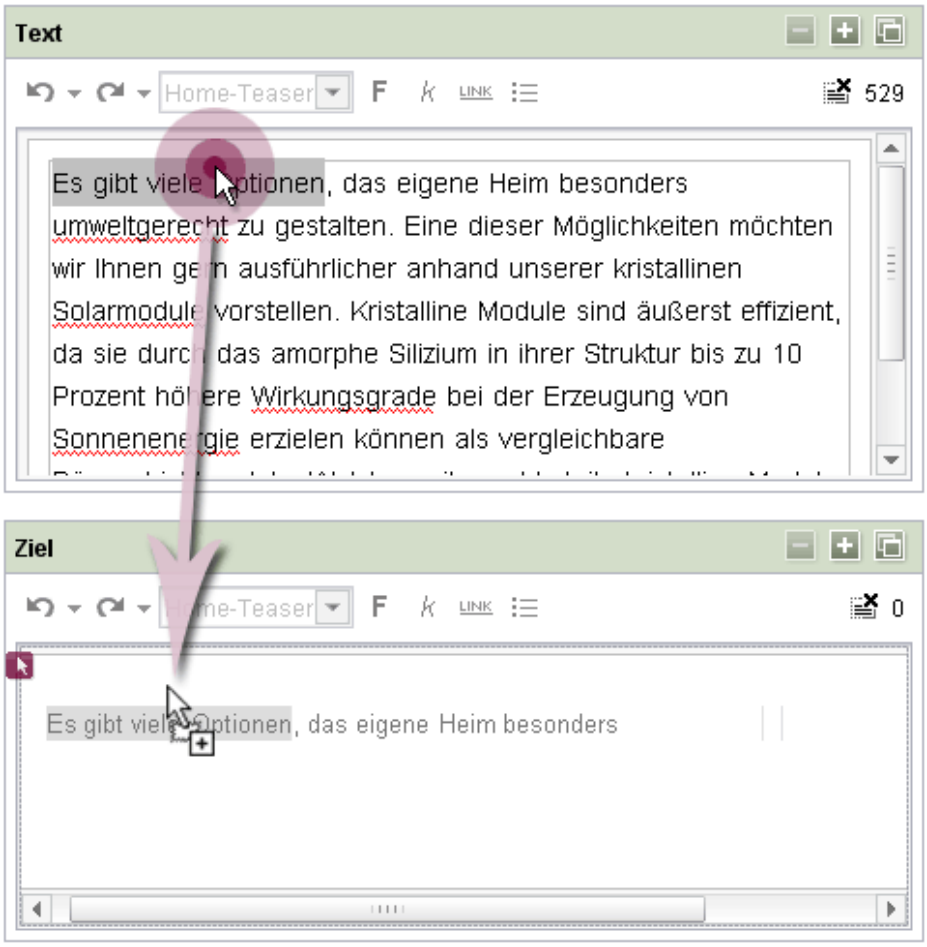


Figure 6-22: Drag and drop of content

Text, e.g. in a text field, can be selected with the mouse and then also moved or copied into other text fields or input components. In the case of text from CMS_INPUT_DOM formatting is also moved or copied, provided the configuration allows this. These drag-and-drop operations can also be made using the

- context menu or keyboard shortcuts (Chapter 6.1.11 page 67) or
- clipboard (Chapter 6.1.2.7 page 48)

Detailed documentation of the drag-and-drop functionalities in the FirstSpirit JavaClient is provided in the FirstSpirit Manual for Editors (JavaClient), "Drag and drop Functionalities of FirstSpirit" chapter.

6.1.11 Enhanced functionalities in input components

Several functions, accessible via a context menu, have been added for more convenient working with the input components available in FirstSpirit.

For easy transfer of content, all input components now have a context menu, with which the contents of the input component can be cut, copied and pasted. It can be opened by right-clicking the title or frame of the input component.

Figure 6-23: Context menu on input components

Alternatively, these functions can also be performed using the keyboard shortcuts

- <Ctrl> + X (Cut)
- <Ctrl> + C (Copy)
- <Ctrl> + V (Paste)

If values cannot be copied from one input component to another due to incompatibilities, the "Paste (Ctrl+V)" entry is disabled.

In the case of text-based input components such as CMS_INPUT_TEXTAREA, CMS_INPUT_TEXT, CMS_INPUT_NUMBER, CMS_INPUT_DATE and CMS_INPUT_LINK, the following functions are now also available, by right-clicking in the text field:

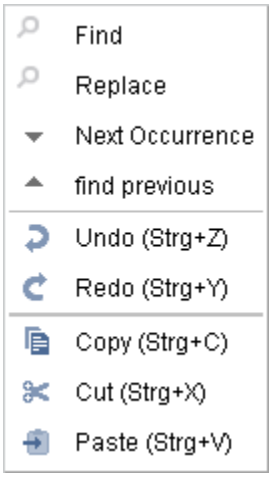


Figure 6-24: Extended context menu on input components

- **Find:** Opens a dialog, with which the input component can be searched for a specific text.
- **Replace:** Opens a dialog, with which the input component can be searched for a specific text and this text can be replaced.
- **Next Occurrence:** If a search for a specific text has been formed, this entry can be used to switch to the next search hit in the input component. The search does not end at the end of the input component.
- **find previous:** If a search for a specific text has been formed, this entry can be used to switch to the previous search hit in the input component. The search does not end at the start of the input component.
- **Undo:** Resets all changes in the input component since Edit mode was activated.
- **Redo:** Restores all changes reset using Undo.
- **Paste:** Inserts content from the clipboard in the cursor position.
- **Copy:** Copies the highlighted content into the clipboard.
- **Cut:** Removes the highlighted content and copies it into the clipboard.

In the DOM editor (CMS_INPUT_DOM), the following keyboard shortcuts can also be used in FirstSpirit 5.0 (provided the corresponding functions are available in the respective input component):

- <Ctrl> + B: Bold formatting
- <Ctrl> + I: Italic formatting
- <Ctrl> + <Shift> + R: Dialog for inserting a reference
- <Ctrl> + <Shift> + L: Dialog for inserting lists

- <Ctrl> + <Shift> + P: Focus on selection list with section formats, selection of the required section format template using the Cursor keys
- <Ctrl> + <Shift> + T: Focus on selection list with character formats, selection of the required section format template using the Cursor keys

6.1.12 CMS_INPUT_LINK: New display of the link input

A new form of representation has been introduced in Version 5.0 for the input component for entering a link (CMS_INPUT_LINK) in JavaClient: Whereas before dialogs had to be opened to display the saved information on a link (e.g. link target and link text):

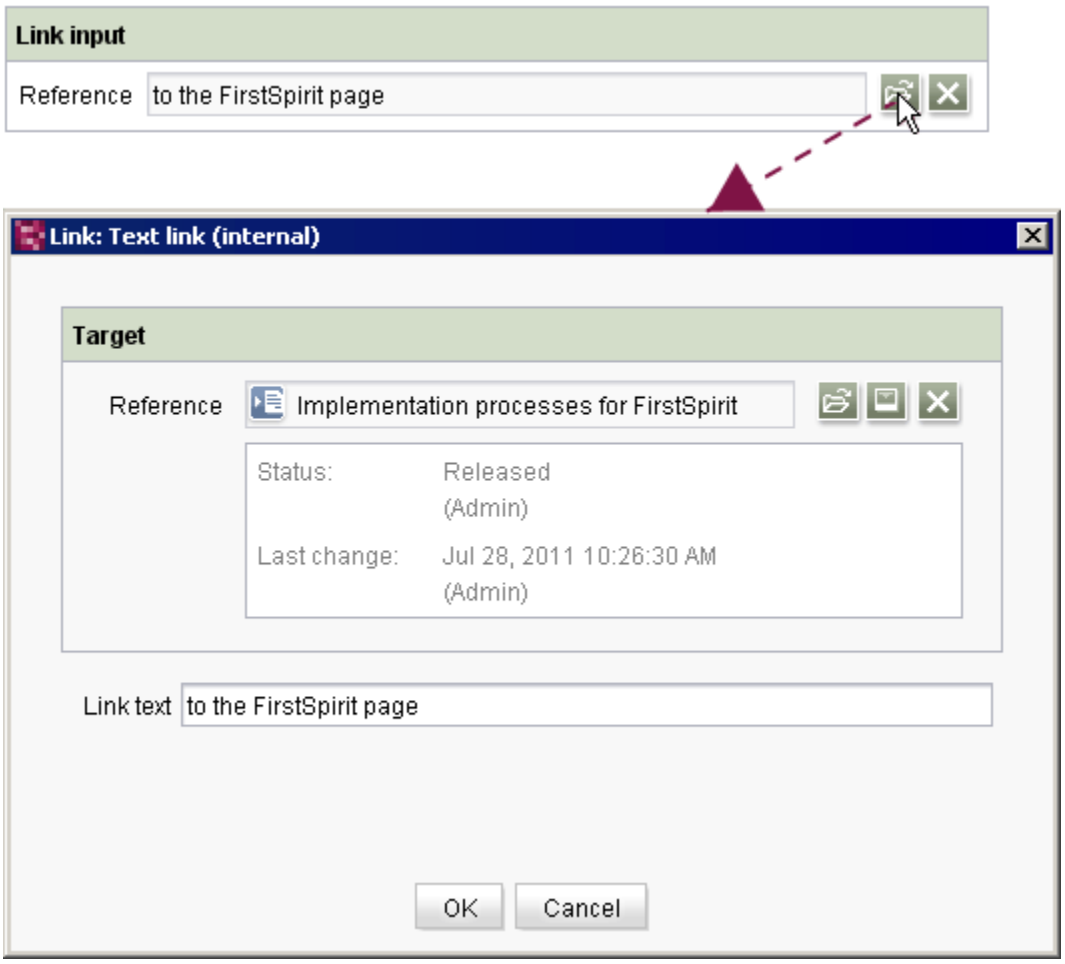


Figure 6-25: CMS_INPUT_LINK (Dialog)

they can now be displayed directly in the section, in the page or in the data record:

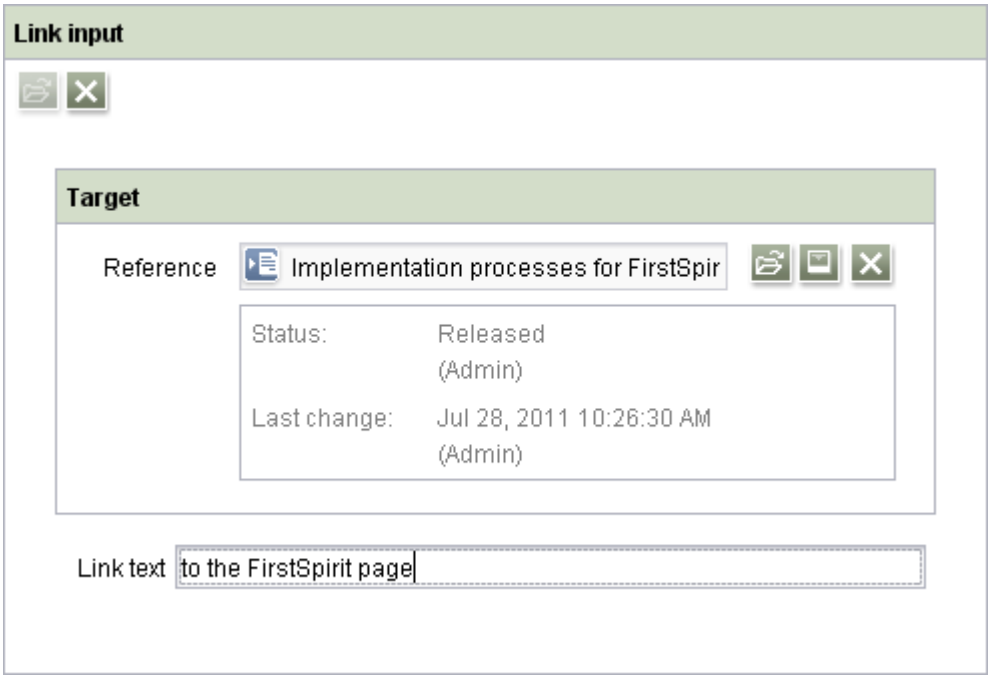


Figure 6-26: CMS_INPUT_LINK (Inline)

The X icon must be pressed first before another link type can be selected.

6.1.13 New display of data records (FS_DATASET)

A new form of display has been introduced for the input component for the selection of data records (FS_DATASET) in JavaClient in version 5.0:

While by default all information of a data record are shown in text fields in the input component:

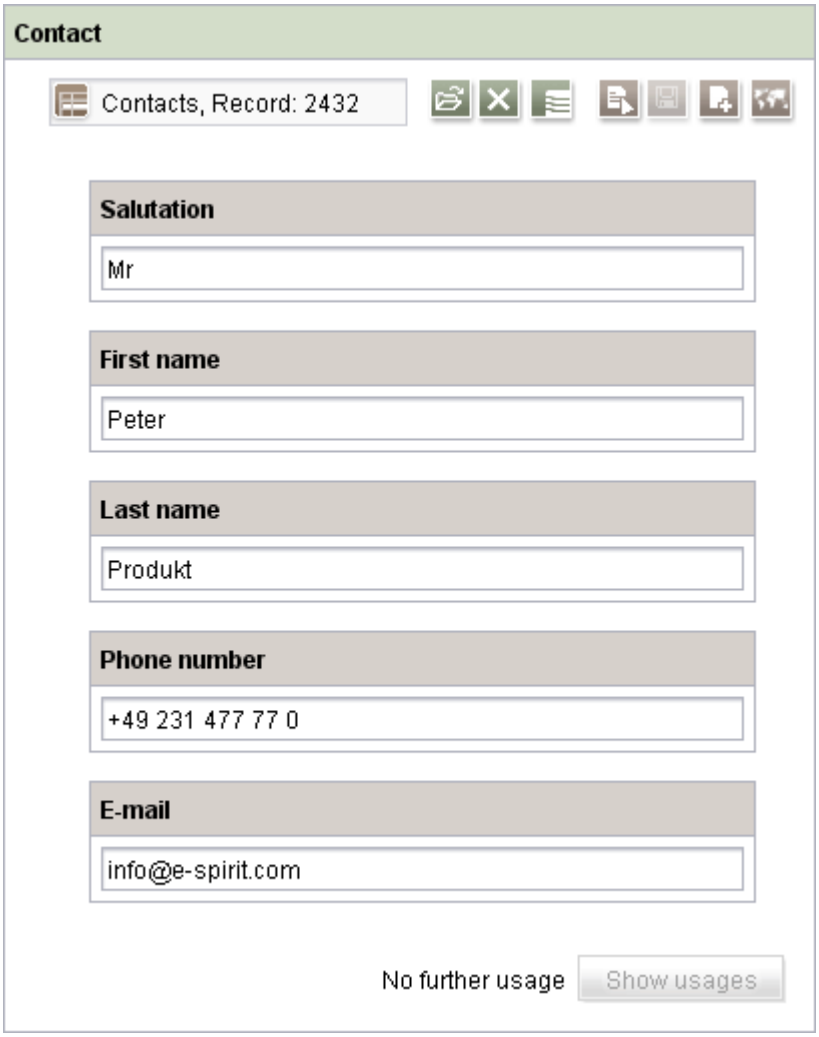


Figure 6-27: FS_DATASET, default display

the information can now be displayed shortened if configured by the template developer, e.g. only with the name of the data source, the ID of the data record and the path:

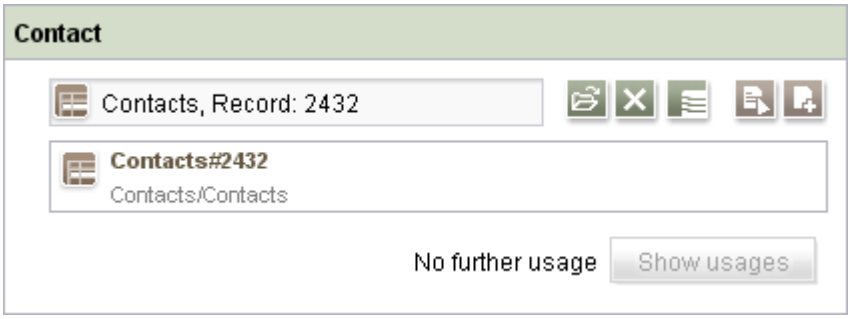


Figure 6-28: FS_DATASET, shortened display of the data record

or with selected information within two rows:

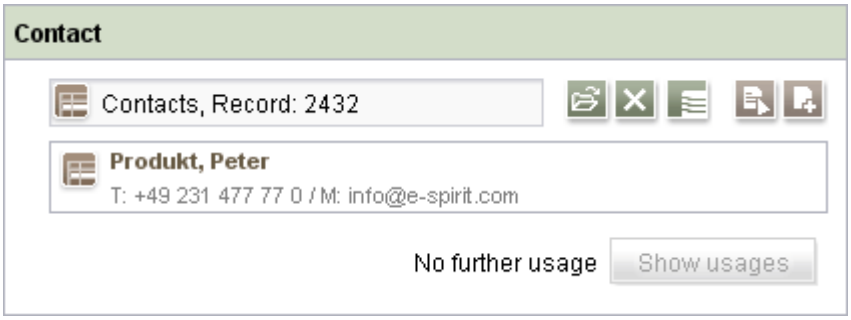


Figure 6-29: FS_DATASET, shortened display of the data record

6.1.14 Optimized management and display of media

6.1.14.1 Exif data

Images and files with different formats can be filed and managed in the FirstSpirit Media Store. With FirstSpirit 5.0, new infrastructure has been introduced for mime type recognition and an Exif library ("Exchangeable Image File Format"). The objective is to categorize media and to be able to more flexibly respond to files with the wrong file extension.

Depending on the configuration set by the project developer, this Exif data of imported pictures can be displayed on the "Metadata" tab of a picture and if applicable can also be changed, e.g. camera manufacturer and model, date and time taken, focal length, exposure time).

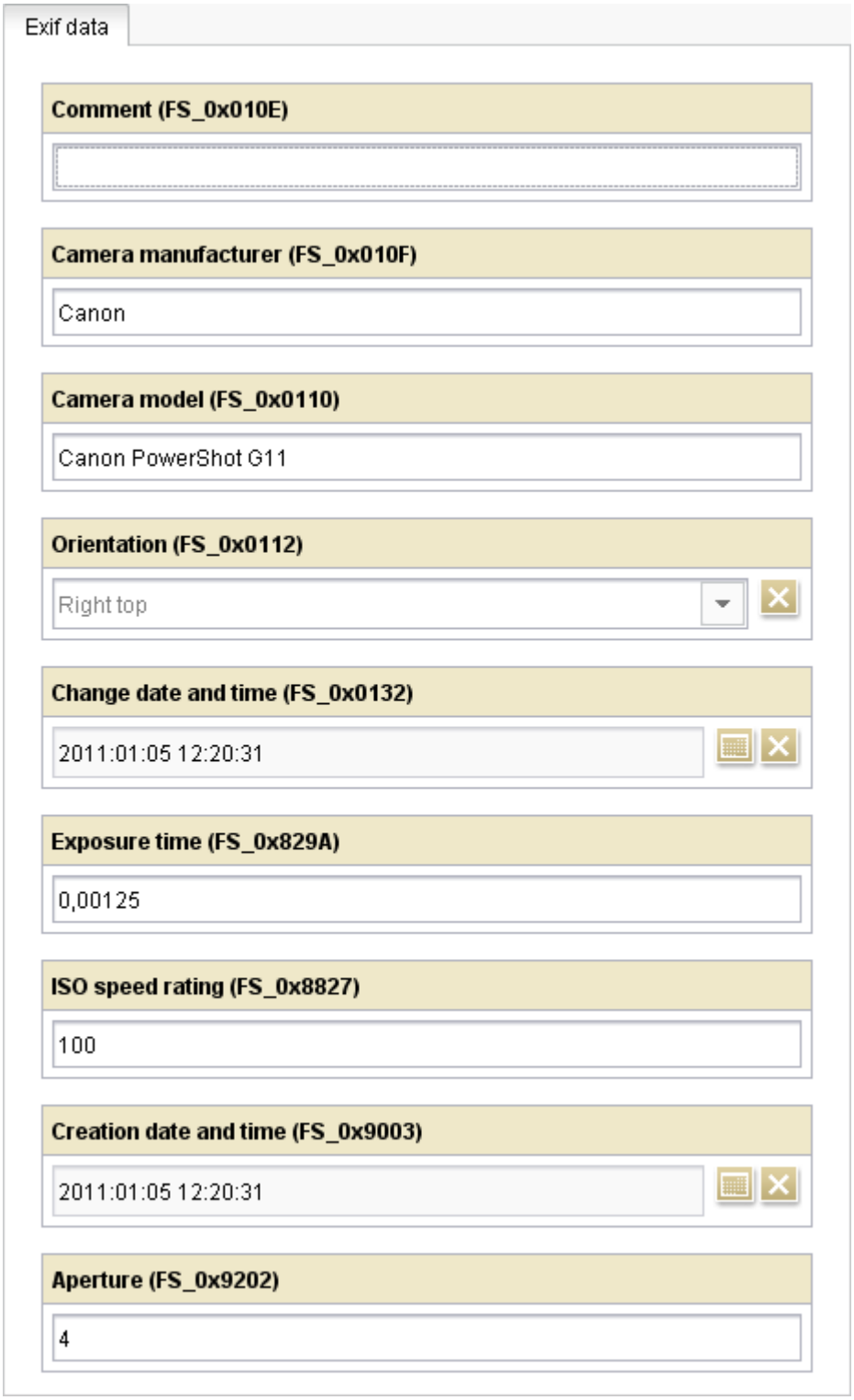


Figure 6-30: Exif data of a picture in the Media Store

Which data is displayed or entered depends on the configuration set by the project developer.

In addition, media in the Media Store can now be automatically defined by the system according to their MIME type ("Multipurpose Internet Mail Extensions"). This means that even if a file has the wrong file name extension when uploaded into the Media Store (e.g. "*.jpg" instead of "*.png"), the format is nevertheless correctly identified ("*.png") and is saved with the picture. It is also possible to filter by these MIME types in the search (see Chapter 6.1.2.4 page 43).

6.1.14.2 Display of media in the Media Store

Whereas until now thumbnails (reduced-size view) of pictures only could be displayed, it is now possible to show **thumbnails** of files such as PDF or Microsoft Word. This depends on the project developer's settings.

Media on the "Overview" tab (folder level) are now framed as long as the mouse pointer is held above them. Selections are permanently shown with a frame.

An enhanced **tooltip** is now displayed for media already on the tab "Overview" (folder level) for a quicker overview of the data of a medium. This tooltip contains the following information (if available):

- file type icon
- path
- name
- description
- width, height
- file extension
- size of the file
- type of the file

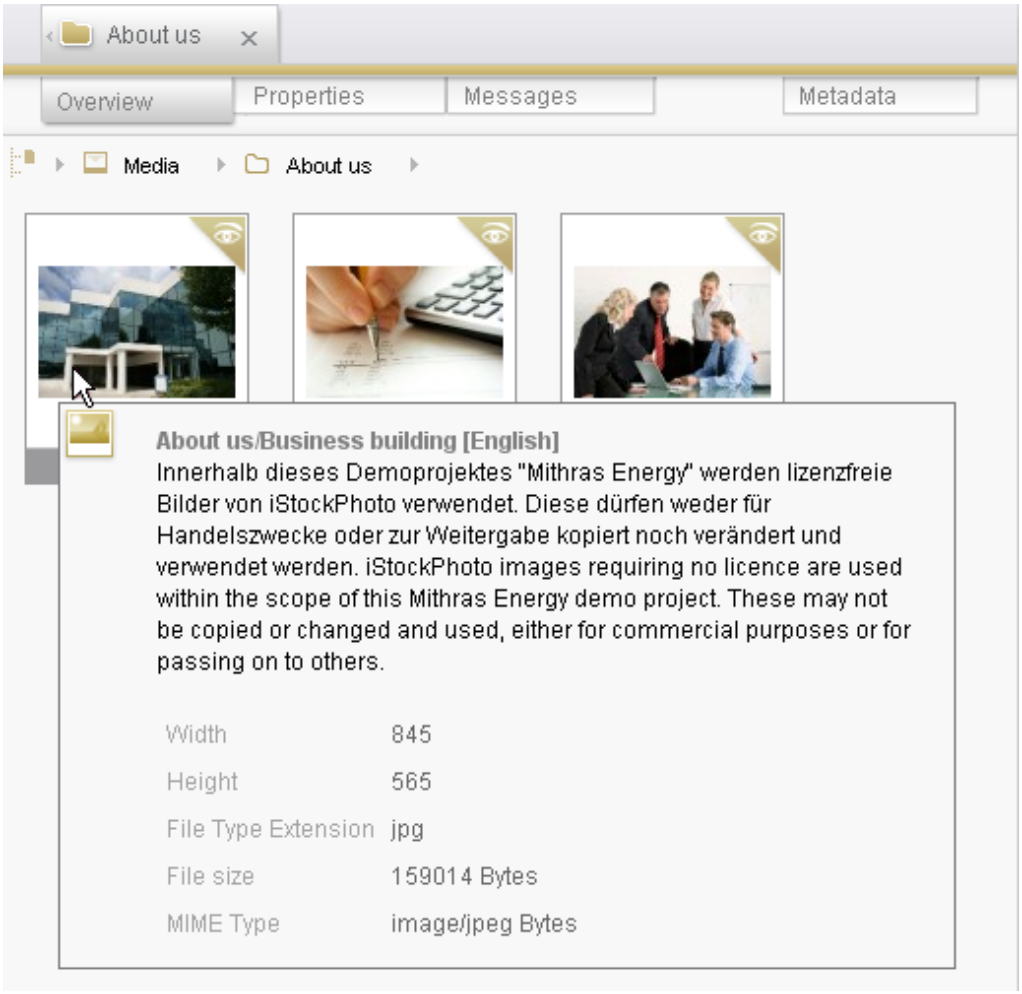


Figure 6-31: Tooltips of media in the overview

6.1.15 URL generation management (search engine optimization)

The functions which are described in the following require in-depth knowledge about the generation function in FirstSpirit and URLs and aims for this reason rather project administrators. WARNING: When modifying the strategy of creating URLs it can become necessary (especially in an existing project) to adjust accordingly downstream processes (e.g. the publication). See also Chapter 8.5.4 page 135.

The naming of objects in FirstSpirit makes a strict distinction between display names (not unique, optionally maintainable mutlilingually, with Unicode support) and reference names (unique within the namespace, restricted to letters and numbers, i.e. no Unicode support). While the display names are relevant for the editing tasks and can be changed by the editor at any time, the

reference names are normally only required by the template developer and for intrasystem actions and cannot be changed (or only with great effort).

This two-level naming has proven its worth in practical application, but means that reference names must be relied upon at certain junctures. URLs, for example, may not include Unicode characters as per specification. These structures are hence based on reference names in FirstSpirit and cannot be fully influenced by the editor for this reason.

As a consequence, FirstSpirit Version 5.0 now implements an option for exerting a greater influence on URLs than heretofore, e.g. for search engine optimization.

From FirstSpirit Version 5.0, the following concepts will apply to the generation of URLs:

- **SEO URLs**
  This concept allows URLs, which were previously formed monolingually in FirstSpirit from the folder structure of the project and file names of page references, to be freely defined for every node (menu levels and page references). This not only provides the option of renaming folders / directories and files, but also allows the creation of a directory structure for the web server that completely departs from the website structure and/or project. This means that this

```
http://domain.de/de/events/cebit.html    and
http://domain.de/en/events/cebit.html
```

  can be turned into this

```
http://domain.de/veranstaltungen/cebit.html  and
http://domain.de/events/cebit.html
```

- **Short URLs**
  Short URLs are brief, easily remembered, "expressive" URLs, e.g. for so-called "landing pages". The latter term stands for specifically created individual webpages where a specific topic or offer is presented in a compact manner and which are optimized for a specific target group and the subject of the page. These are often the target pages of linked adverts in other webpages. They are mostly inaccessible via the website's navigation. Short URLs are generated in addition to the "normal" URLs.
  This URL

```
http://mithrasenergy.com/content/de/ueberuns/unternehmen/Unternehmen.html
```

  would for example be turned into this URL

```
http://mithrasenergy.com/company
```

  Every page reference can be provided with several alternative short URLs.

The JavaClient now offers the node "URL Settings" in its "Global Store" for this.

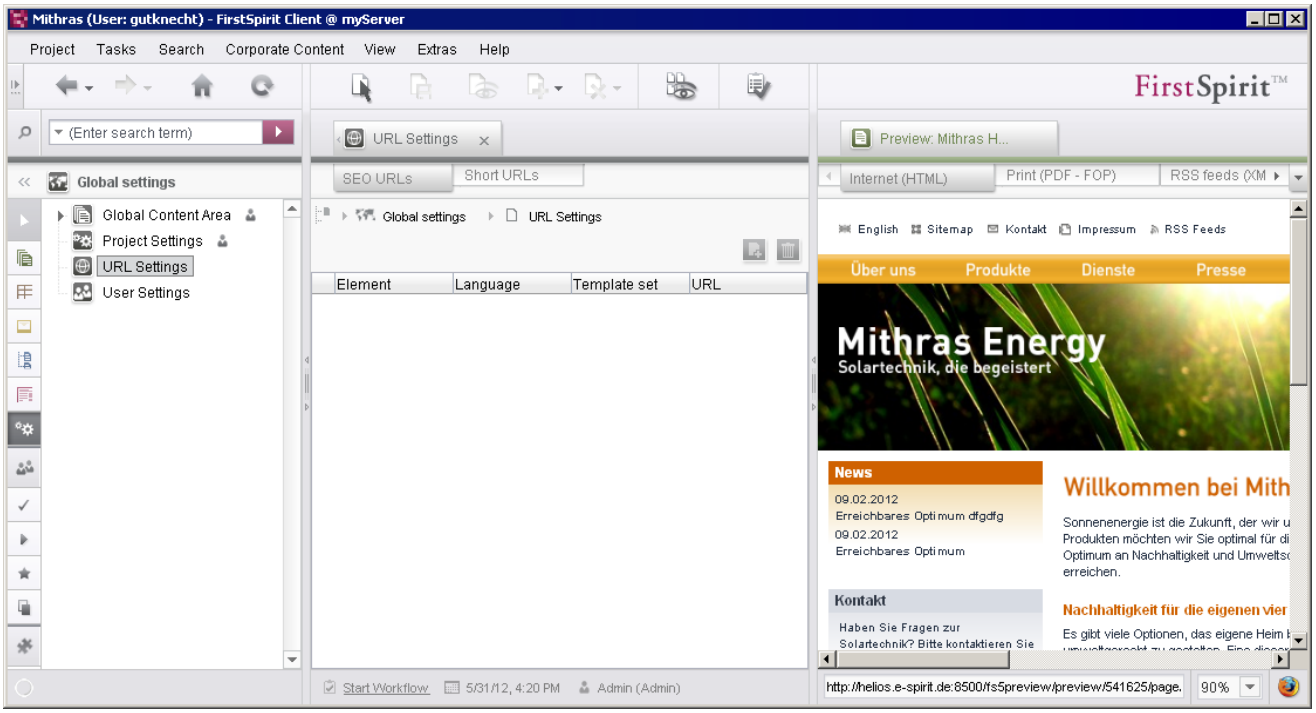


Figure 6-32: Global Store – overview with URL settings

The two hanging tabs

- "SEO URLs" (see Chapter 6.1.15.1 on page 78) and
- "Short URLs" (see Chapter 6.1.15.2 on page 83)

available here allow the URLs to be influenced. Both of these overviews can contain several entries.

Project administrator rights are required for editing the tabs.

Element: Shows the path and element for which a processed URL is available. This can be a page reference or a menu level.

Language: Shows the language the processed URL applies to

Template set: Shows the channel the processed URL applies to

URL: Shows the URL created for the respective element in the generation process

6.1.15.1 "SEO URLs" tab

Add, This icon serves to select a menu level or page reference from the Site Store that a URL is to be manually specified for. The first stage opens a selection dialog:

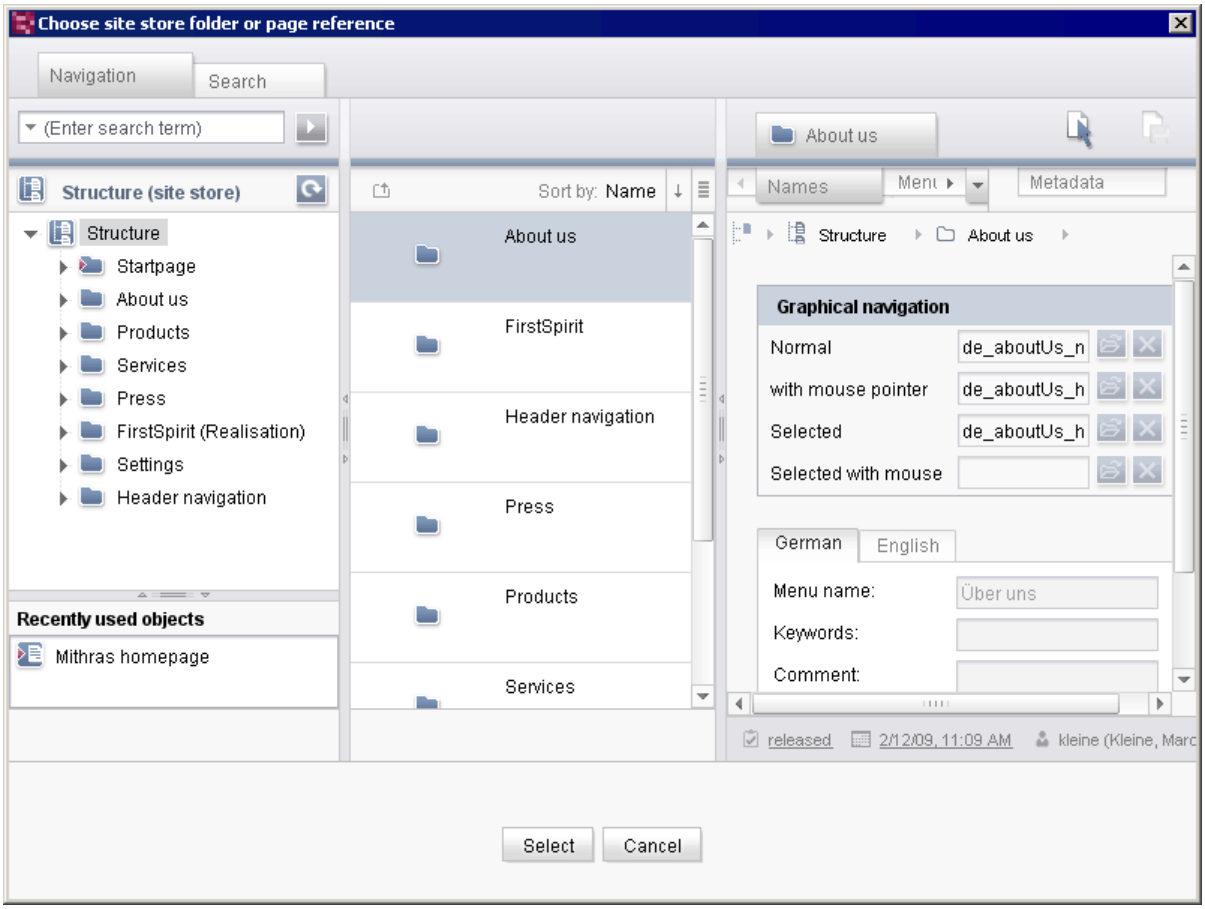


Figure 6-33: Editing URLs – selecting a menu level or page reference

In this dialog one can select the menu levels (see "Application examples for menu level URLs") or page references (see "Application examples for page reference URLs") the URLs are to be defined for. As menu levels are converted into folders during the generation process and page references into files in a directory structure that is subsequently transferred to a web server, for example, the directory / folder path can be influenced for menu levels using the "URL settings" function, and for page references also via the file name. In the process, subdirectories are created / shown by inserted slashes ("/").

More information on working with this dialog can be found in the *FirstSpirit Handbook for Editors*

(JavaClient), Chapter "Search in selection dialogs ".

Clicking "Select" opens the following dialog:

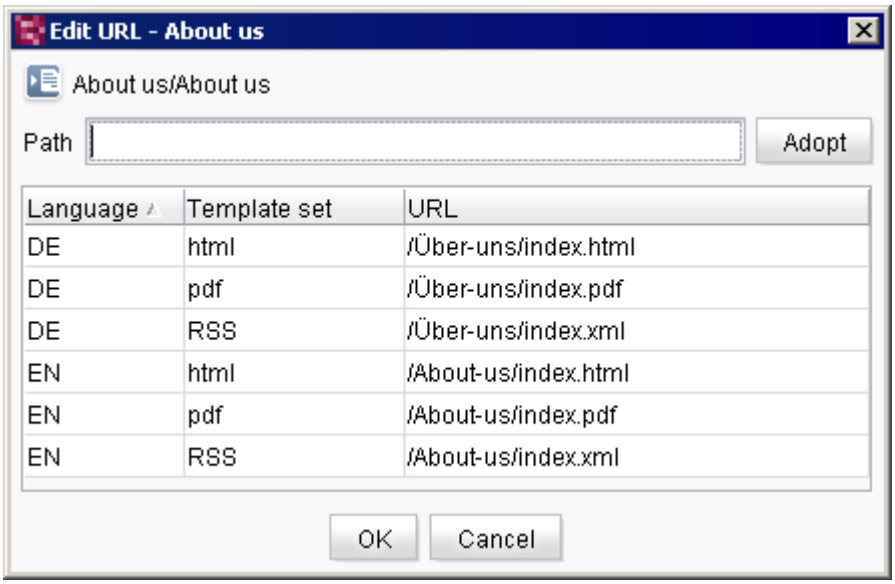


Figure 6-34: Editing URLs

In this dialog, the directory paths the URLs are derived from for the selected node can be changed for the languages and template sets available in the project.

The table can be navigated by keyboard shortcuts as follows:

- **ENTER** or **down arrow key**: Go to next line / one down
- **ENTER + SHIFT** or **up arrow key**: Go to previous line / one up
- **F2:** Enter input in the "URL" field in the selected line.

If the "URL Settings" function is invoked on a menu level, only the names of paths/directories can be influenced. These changes will affect all subordinate objects that are not provided with an SEO URL definition of their own. If the function is invoked in page references, a path with a unique file name can be defined for every language and template set.

URL: The "URL" field in the table initially shows the currently saved URL of the node for every language and every template set. The entries can be sorted by clicking the column header. The URLs can be edited by double-clicking the fields or pressing ENTER + CTRL. Slashes at the beginning of entries are automatically inserted for both menu levels and page references.
Each URL (path plus file name) may meanwhile only be defined once within a project.

Changed entries are shown in italics. If the object concerned is a so-called multi-page (page with a data record from a data source, "content projection"), the URL setting will always only apply to the first page.
The field can also be left blank.

Path specifications can be applied to all languages and template sets by entering them in the "URL" field above the table and adopting them for all the entries in the table by clicking "Adopt". If the "URL" field is left blank, clicking "Adopt" will delete all entries in the "URL" column. Only filled entries will be applied. If the path is only to be applied to specific entries, the corresponding entries must be selected beforehand; several entries can be selected by pressing CTRL or SHIFT at the same time. The path names can also be adjusted later on. In doing so, a slash will be automatically placed at the beginning of entries.

The entries provided here are also used for building the directory structure where the results of the generation process are filed (see Chapter 8.5.1 on page 125).

Application examples for menu level URLs

The options available for menu levels for example include the possibility of

- assigning **other names** than the display names in the tree structure. This

```
../Startpage
```

  could for example become this

```
../Welcome
```

  by entering "/Welcome" in the "Startpage" node.

- shortening **paths**.
  This

```
../Pressemitteilungen/Presse/Mithras Energy erhält Solarpreis der Stadt
Sonningen.html
```

  could for example become this

```
../Pressemitteilungen/Mithras Energy erhält Solarpreis der Stadt
Sonningen.html
```

  by defining "/Pressemitteilungen" in the "Presse" node.

- or of adding **additional sub-directories**:

```
../Pressemitteilungen/PDF
```

```
../Pressemitteilungen/RSS
```

Application examples for page reference URLs

These examples relate to the "Mithras Energy" demo project, "Advanced URLs" generation mode (see Chapter 8.5.4 on page 135).

An `index.*` file with the file extension of the respective template set (`*.html`, `*.pdf`, `*.xml` etc.) will be generated for page references by default.

In the case of page references

- different (file) names can be assigned by changing the file name,
- the path to a file can be shortened or expanded by removing or adding directories (with "/").

The relevant file extensions must be assigned in the process.

OK Clicking this button applies the settings to the overview (see Figure 6-32).
If URLs (path plus file name) have been assigned several times over (other languages and/or other template sets), a corresponding message will be displayed ("The URL '...' is used repeatedly!").
If a manually entered URL is already being used for another node in the project, the duplicate URLs are shown in red in the overview. When saving or exiting the editing mode using CTRL + S or CTRL + E, respectively, or the corresponding icons in the JavaClient tool bar or the context menu option "Editing on/off", the message "Please correct duplicate URLs first.' will be displayed.

The icon will display page references and their attendant settings in the overview (Figure 6-32).

URLs that are already provided in the overview can be changed by double-clicking the respective field in the "URL" column. If an element is not yet provided in the language and/or template set in the overview the URL is to be changed for, the entry can be added for the desired node using the

 icon (see top of this Chapter).

 Delete, This icon serves to delete a line of an entry from the list and hence reset the manual URL settings for this element.

URL settings and/or changes must be saved (CTRL + S or "Save" icon in the JavaClient menu bar) or the editing mode of the node exited again (CTRL + E, corresponding icon in the JavaClient menu bar, or context menu option "Editing on/off".

 *The changes made here will only be included in the generation process if a corresponding schedule is carried out. This is normally done by the project administrator (see also Chapter 8.5 on page 125).*

 *The automatically assigned URLs that are saved in the node can be viewed in the object information (using ALT + P or context menu "Extras" / "Show properties").*

6.1.15.2 "Short URLs" tab

Add, This icon serves to select a page reference a short URL is to be defined for from the Site Store. The first stage opens the following dialog:

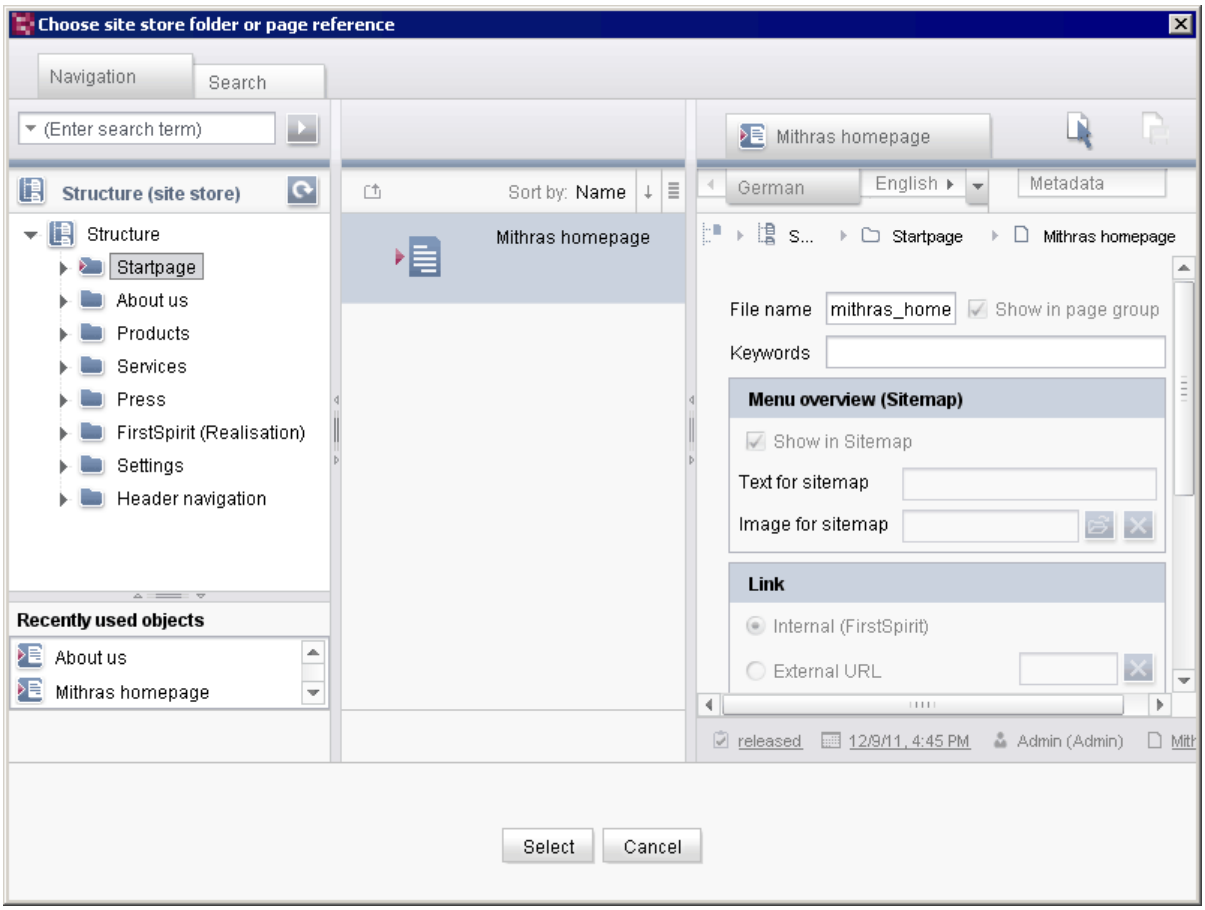


Figure 6-35: Defining short URLs – select page reference

Several alternative short URLs can be defined for each page reference.

The definition process is the same as for SEO URLs (see Chapter 6.1.15.1 on page 78), except that no menu levels can be selected for short URLs.

Application examples

A short URL can for example be based on a **product name** or **product category** that customers and prospects are directly looking for, e.g.

```
www.mithras-energy.com/storageunit
```

or on a **specific topic** circulated by promotional materials, e.g.

```
www.mithras-energy.com/cebit
```

or

```
www.mithras-energy.com/offer
```

To create a short URL for a **product page**, for example, that is based on a content projection (page with a data record from a data source), a page with a data source-based section must be created in the Page Store. What is required is a data base query whose result is exactly the data record that is to be displayed in the page. This data base query can then be selected in this page's instance in the Site Store (page reference with "Content" tab). The desired short URL for this page reference can then be filed in the URL settings:

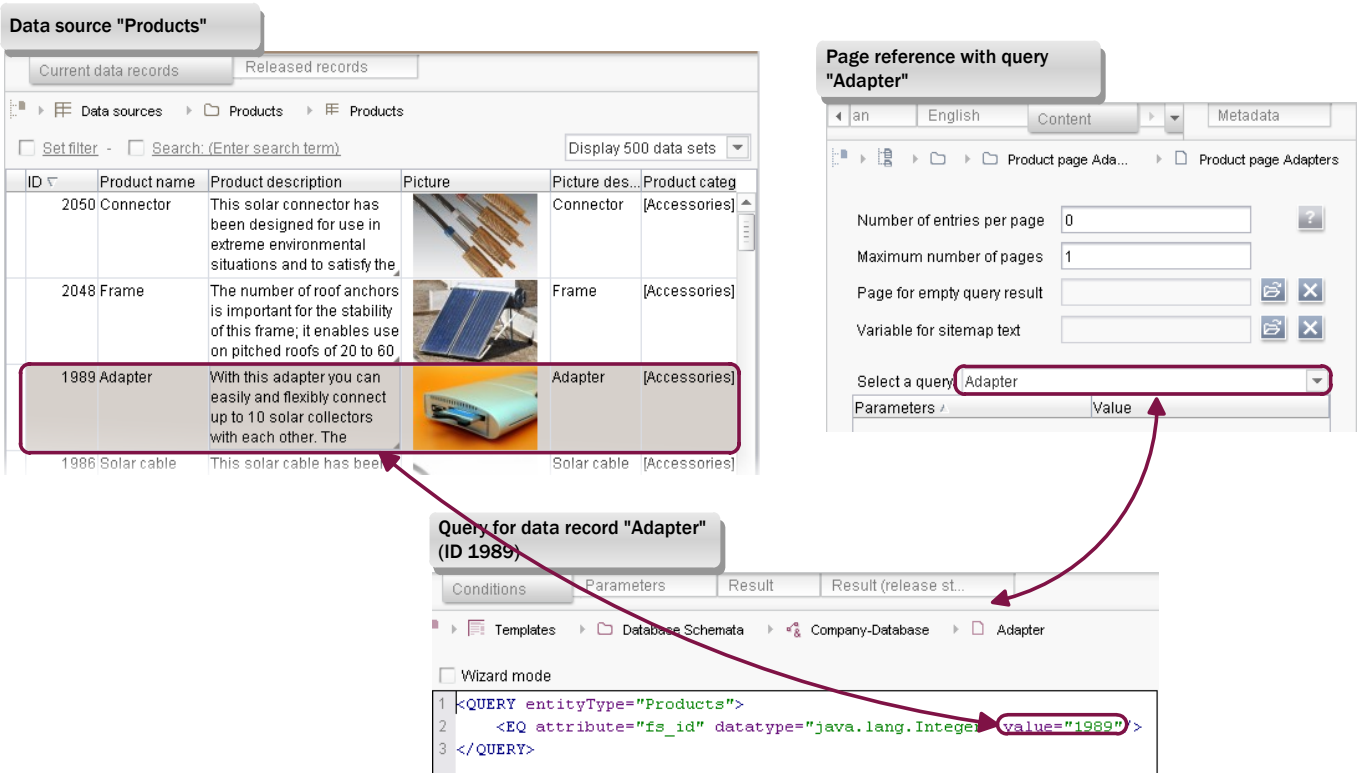


Figure 6-36: Page reference for a product page

OK This icon adopts the settings in the overview (see Figure 6-32).

If URLs (path plus file name) have been assigned several times over (other languages and/or other template sets), a corresponding message will be displayed ("The URL '...' is used repeatedly!").

If a manually entered URL is already being used for another node in the project, the duplicate

URLs are shown in red in the overview. When saving or exiting the editing mode using CTRL + S or CTRL + E, respectively, or the corresponding icons in the JavaClient tool bar or the context menu option "Editing on/off", the message "Please correct duplicate URLs first.' will be displayed.

Several different URLs can be defined for each page reference, however.

The  icon will display page references and their attendant settings in the overview (Figure 6-32).

URLs that are already provided in the overview can be changed by double-clicking the respective field in the "URL" column. If an element is not yet provided in the language and/or template set in the overview the URL is to be changed for, the entry can be added for the desired node using the  icon (see top of this Chapter).

 Delete, This icon serves to delete a line of an entry from the list and hence reset the manual URL settings for this element.

URL settings and/or changes must be saved (CTRL + S or "Save" icon in the JavaClient menu bar) or the editing mode of the node exited again (CTRL + E, corresponding icon in the JavaClient menu bar, or context menu option "Editing on/off".

 *The changes made here will only be included in the generation process if a corresponding schedule is carried out (see Chapter 8.5 on page 125).*

6.1.16 Content Transport ("Add to Content Transport feature")

The function "Template update" familiar from version 4.x has been reimplemented in its entirety in Version 5.0 and turned into a new function named "Content Transport" that offers a significantly expanded range of options with a focus on the realization of scenarios requiring the application of development, test and productive systems (DQP).

Content Transport features can be created and edited in the left area of the JavaClient using the icon  .

You will find a new entry "Add to Content Transport feature" in the context menu of nodes which can be used for a transport for adding objects from the tree structure to a Content Transport feature.

For detailed information about functionality and use please see module documentation FirstSpirit CoporateContent.

6.1.17 Comparative view in the integrated preview

With FirstSpirit 5.0 – depending on the settings made by the project developer / administrator – it is possible to open a node in the Page or Content Store or in the Global Content Area in the integrated preview. In this way, sections, pages and data records can be permanently displayed in the preview their form view, in order to compare the contents of their input components with others (e.g. when revising pages, sections or data records) or to copy content from these input components into input components located in the workspace.

To do this, the "Show form in new tab" entry of the "Plug-ins" context menu is opened on the node whose view is to be opened in the integrated preview (sections and pages in the Page Store and in the Global Content Areas, data records, content sources). A new tab opens in the integrated preview, which shows the input components of the selected node. Content cannot be edited on this tab.

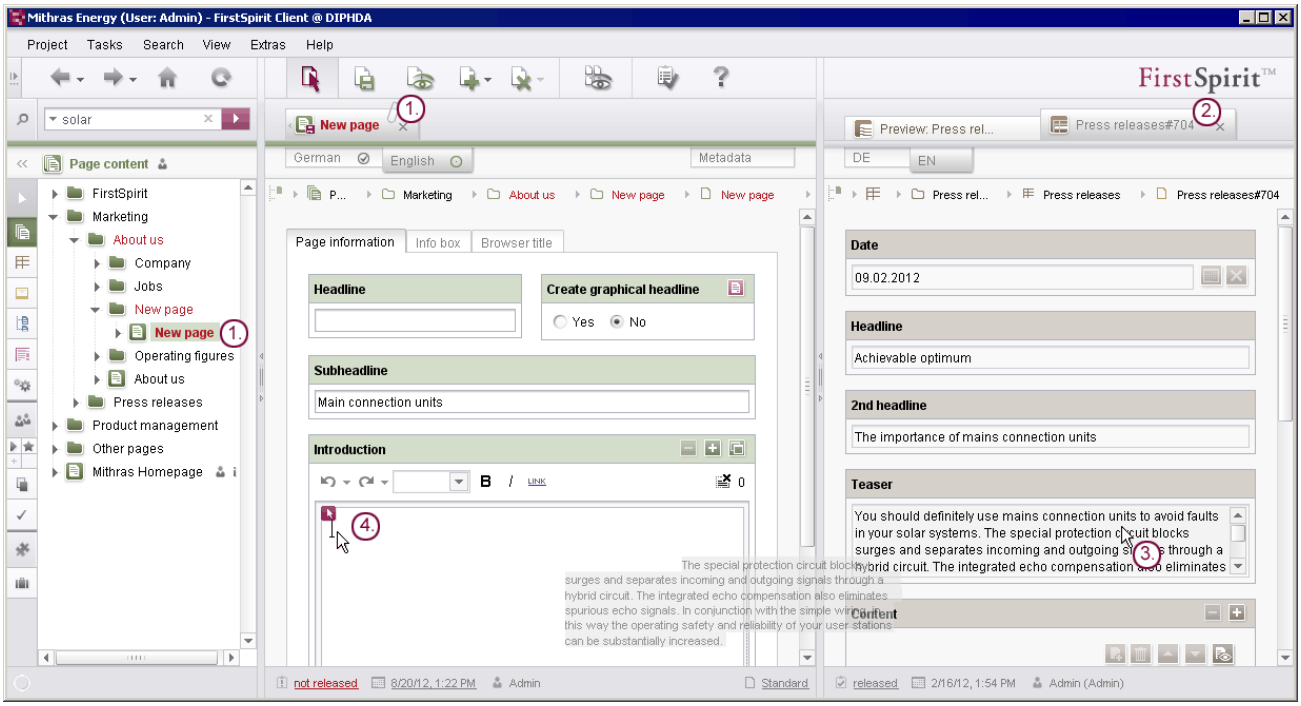


Figure 6-37: Opening a data record in the integrated preview

In the example shown, a page with its available components has fist been placed in Edit mode in the workspace (1.). The contents of a data record, which is displayed via the script on the data record in the integrated preview, can be used for the input of content (2.). Using Copy & Paste (3.), content from the input components available there can be copied into the input components in the workspace (4.). A feasible option would also be to show the contents of an element in two different languages, quasi next to each other.

The tab opened in this way remains open until it is closed using the X on the right-hand edge of the tab or the context menu entries "Close preview area" or "Close other preview areas".

In Version 5.0 and higher, the contents of several workspaces can also be compared with each other using the enhanced workspace view ("TabView") (see Chapter 6.1.18 page 87).

6.1.18 Expanded work area view ("TabView")

Depending on the settings in the “View” / “Work Areas” menu and the individual work modes, it is possible to open with the time plenty of work-area registers in JavaClient. For the simpler operation of the work areas, they can now be presented in a clearer view and controlled more

comfortably.

 With this icon in the JavaClient toolbar, an overview of the work areas can be displayed in the AppCenter area:

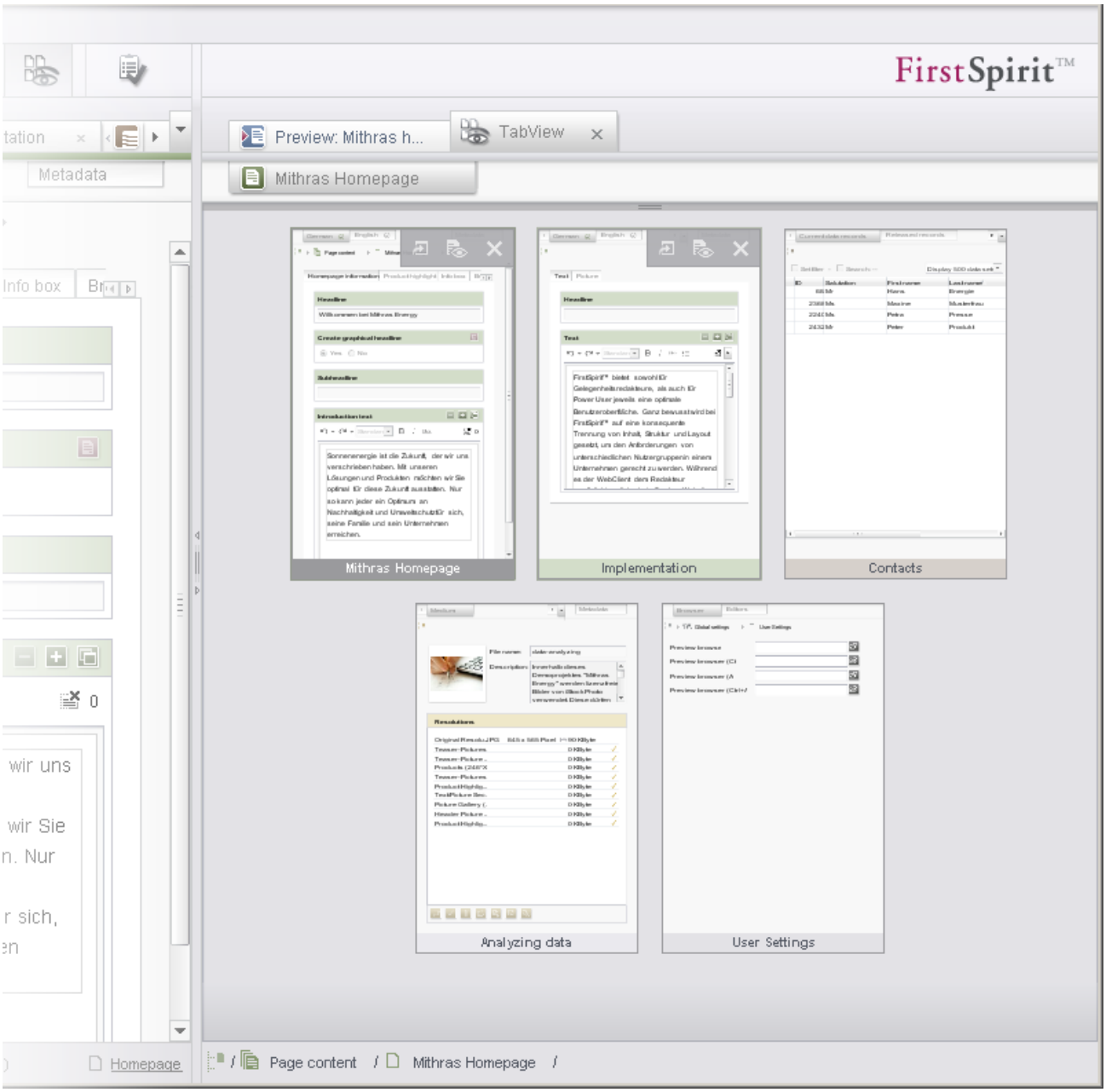


Figure 6-38: Work areas overview in the AppCenter area

This view shows all currently opened work areas in a scaled-down preview. The display name of the object is shown under the preview. The currently active work area is marked with a frame.

The corresponding path is shown in the status bar. It can be used also for changing to other nodes.

By clicking on a preview display, a quick change to the desired work area in the middle region of the JavaClients can be made in order to be able to process contents there. To view quickly the contents of a given work area, an enlarged view can be opened by means of Ctrl + mouse click (Figure 6-40) and with another Ctrl + mouse click a change back to the initial view can be made (Figure 6-41).

The work areas within the AppCenter area can be moved by drag & drop and thus the order of the associated registers in the middle region of JavaClient can be changed.

With the help of the function bar at the top side of the AppCenter area, the display of the work areas can be further controlled. The bar is activated when the mouse pointer is held over the top side area:

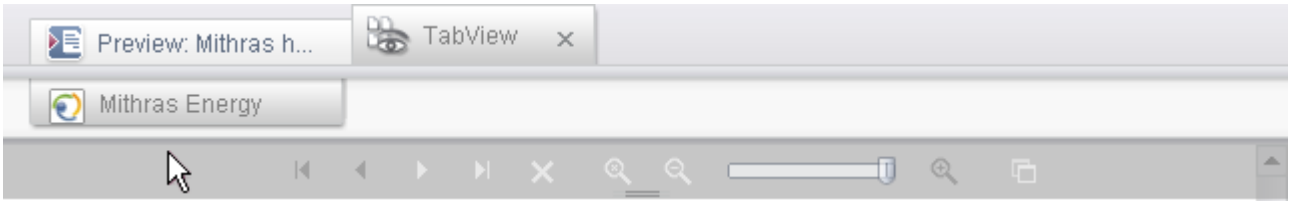


Figure 6-39: Function bar

The arrow icons permit to navigate to the

- first (Alt + A)
- previous (Alt + Q)
- next (Alt + W)
- last (Alt + S)

work area.

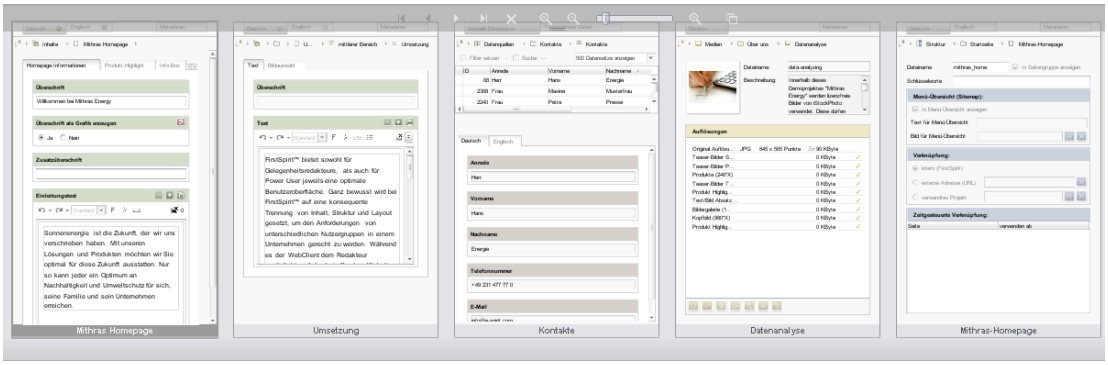
In addition, the following functions are available:

- close active work area (Alt + Esc)

  if all work areas have been closed, they can be redisplayed with the icon

- activate the right work area of the currently active work area (Ctrl + Tab)

  for this purpose, the work areas open first in a horizontal strip:



They can be closed again with Esc or by releasing the Ctrl button.

- Reduce the view (Alt + Minus)
- Enlarge the view (Alt + Plus)
- Reset zoom / normal view (Alt + 0)

The keyboard shortcut Ctrl + Tab or Ctrl + Shift + Tab can also be used, independently of the "TabView" mode, for navigation through the opened work areas. This means that it is not necessary to click on the icon when this shortcut is used.

Depending on the content of the work areas (e.g. pages and paragraphs of the content management), they can be opened in normal view in the AppCenter area. In this way, two work areas can be simply placed in a compare view to each other. Thereby the contents in the AppCenter area are available in view mode only and can not be processed! However, contents from a register in the AppCenter area can be used again in the work areas per copy & paste or drag & drop (as regards the drag & drop and copy & paste functionalities in JavaClient, see also Chapter 6.1.10 page 63 and Chapter 6.1.11 page 67).

In addition, the following functions are available in the top right corner of the thumbnail view:

-  Inline view (Alt + V)
  The thumbnail of the active work area is opened/enlarged in the current register of AppCenter:

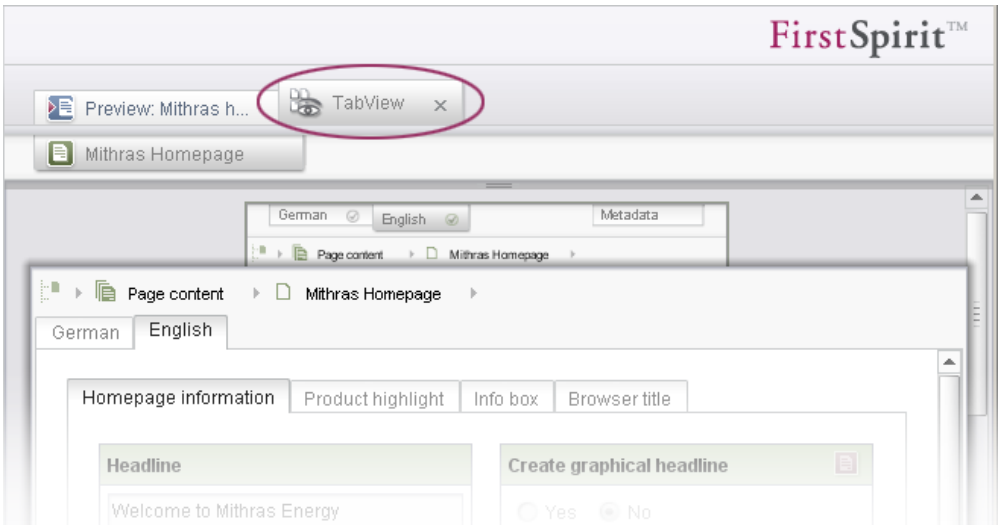


Figure 6-40: Inline view

In the top section of this view, the path to the corresponding object is again displayed. The following functions are available when the mouse pointer is held above the top portion of the work area:

-  Change to the previous inline view (Alt + Q)
-  Change to the next inline view (Alt + W)
-  Close the active inline view (Alt + Esc)

This view can be accessed also with Ctrl + mouse click, and with another Ctrl + mouse click a change to the initial view (Figure 6-38) can be made.

- Open a work area in a new register (Alt + N)
  The thumbnail of the active work area is opened / enlarged in a separate register of AppCenter:

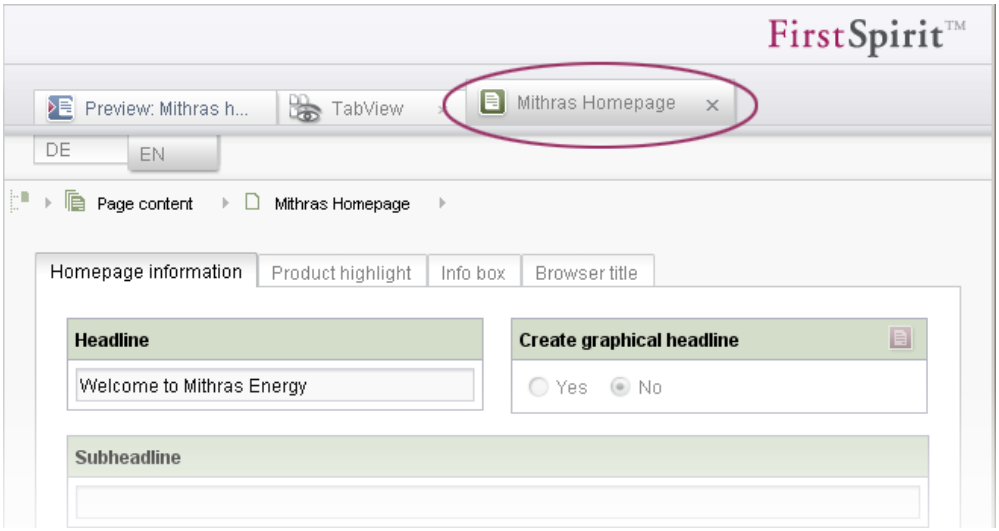


Figure 6-41: New register

Starting from Version 5.0, contents of several work areas can be also positioned next to each other through the compare view in the integrated preview (context menu "Plug-ins" / "Show Form in New Tab") (see Chapter 6.1.17 page 86).

6.2 New/Modified functions in the WebClient

With its WebClient (aka "WebEdit"), FirstSpirit provides editors who only work with content management systems rarely or infrequently with an easy-to-operate and intuitive author interface that enables contents to be edited directly on the webpage in the browser. This way the editor can see immediately what impact the changes will have on the webpage later.

The editing is done on so-called "preview pages" that largely manage without disruptive control elements or pop-up windows, however. The control elements are not permanently visible on the page, but will only be displayed when the editor moves the mouse across an area he or she can edit.

At important junctures, page maintenance is meanwhile supported by assistants (aka "wizards"). These are tasked with guiding the user through the individual maintenance stages, and with informing him or her directly on the inputs made and still required. In an ideal case many aspects will be specified to match here, with the editor only required to attend to a small number of further

inputs in person.

The WebClient in Version 5.0 puts forward a completely new interface. Some functions available in earlier versions of FirstSpirit have meanwhile been discontinued. But this is entirely deliberate in the interest of reduced complexity and not a bug! If special functions are required, project-specific solutions may need to be realized for them. Although the redesign of WebClient 5.0 has been optimized for occasional users, training requirements may still arise.

Some of the new functionalities in WebClient 5.0 are listed below:

- **Page status:** Using the page status in the left upper corner of each page the editor can identificate quickly in which state the current page is (released, modified, needs correction ("Validation issues"), write-protected etc.) and if she requires an action ("Release request"):

Figure 6-42: Page status in WebClient

Here, also older versions of the page can be checked ("Display all changes").

- **Reports:** The so-called Reports represent an important and central source of information within the project. They can be accessed via the right border:

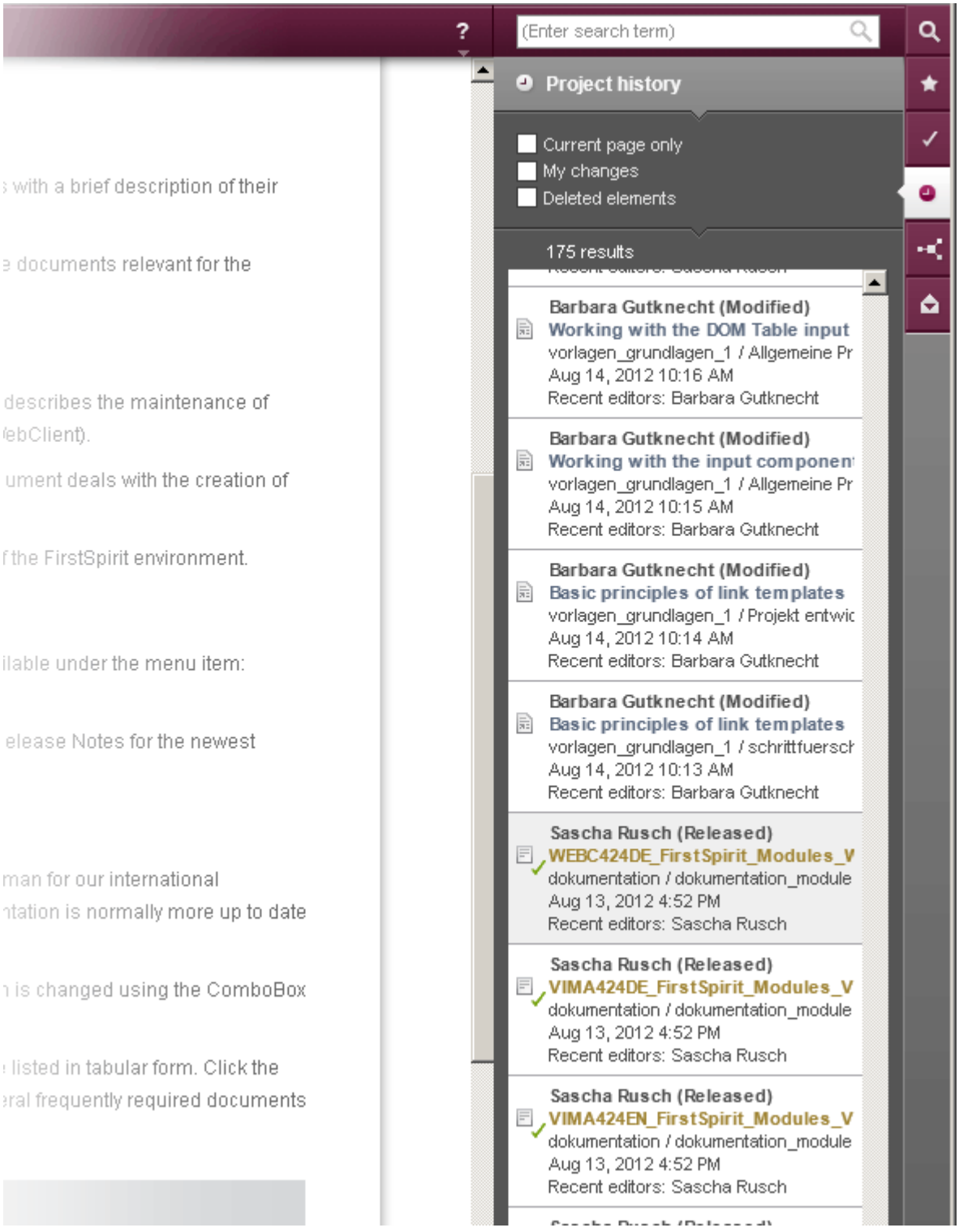


Figure 6-43: Report "Project history"

- **Select image section:** The editor is now able to select an image section in the preview,

directly at the image, that will then be shown on the webpage. He or she can furthermore also rotate and mirror the image to ensure optimal adjustment to the contents and givens of the current page.

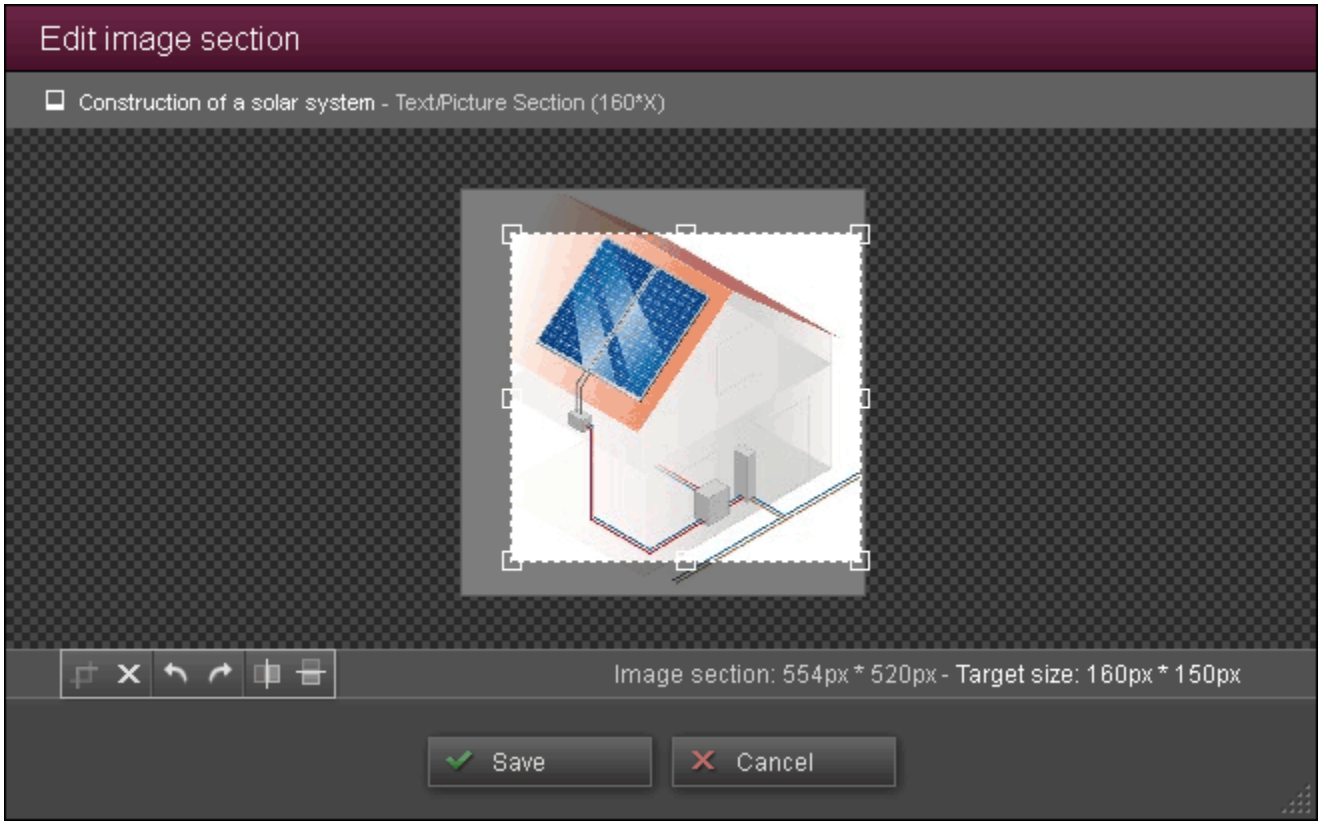


Figure 6-44: Edit image section

- **Drag and drop:** FirstSpirit 5.0 now also introduces drag and drop in the WebClient. This means that media, for example, can now be uploaded to the project by means of drag and drop from the corresponding dialog. In addition to this, objects displayed in the report section (e.g. from searches) can now also be dragged to the preview page. Already existing contents (e.g. images or texts) can furthermore be moved by a mouse click. The areas where objects can be moved to ("dropped") are highlighted correspondingly.
- **Online help:** The user documentation for the WebClient is no longer limited to a PDF format handbook, but now also provided online. In addition, all important icons and buttons have been equipped with tool tips. These will be displayed whenever the mouse stands still for some time at the corresponding locations in the WebClient:
  All viewed tool tips are collected in a list ("Recent Help Topics") in the report section (icon ). From here, the editor can go to the WebClient online documentation directly with a mouse click, where he or she can find further information in an environment of similar or related topics.
  A quick introduction to the WebClient 5.0 is provided by the "WebClient introduction tour" with

brief explanations of its most important control elements. It addresses FirstSpirit novices as well as upgraders from previous versions and JavaClient users and is meant to assist in taking the first hurdles.

- **Inline tables:** Another innovation in Version 5.0 is the possibility of also changing the cell properties in the WebClient using so-called "inline tables".

Below follows a list of functions that have been replaced in comparison with earlier versions of FirstSpirit or implemented in a different manner from that in the JavaClient:

- A number of **input components** which are still provided in the JavaClient are no longer being supported in the WebClient 5.0. The input components that are deprecated in 5.0 should, however, also be replaced in a suitable manner for the JavaClient with Version 5.0. Please see Chapter 7.3.1 on page 101 for more information on the replacements and Chapter 7 on page 99 for more information on the restrictions.
- The WebClient 5.0 will not support projects that do not use **releases** (server and project configuration application, "Options", deactivated option "Use Release function"). The support for projects that do not use releases will also be discontinued for the JavaClient in the 5.x version line.
- WebClient 5.0 no longer supports **language-dependent media** for reasons of usability, and it is also no longer possible to allocate, view or change **reference names**.
- **Navigation** through a project by means of a tree structure is only possible in Version 5.0 if internal references (e.g. media, pages, data records) are selected. The dialog familiar from 4.x versions with a complete tree structure being accessible in the "Extras" menu is no longer available. The WebClient now provides the report section for this on the far right instead, which offers topic- and user-related project views, in addition to navigation through the webpage to be edited using the navigation provided in the preview itself.

Please see Chapter 5.1 on page 34 for further information on new and changed functions, or the WebClient Manual for Editors.

6.3 New/Modified functions in the JavaClient and WebClient

6.3.1 Checking inputs ("Dynamic forms")

With FirstSpirit 5.0, functionalities have been implemented with which entries made in input components can be checked. In this way, stronger interaction with the user is also possible.

Input components now not only accept data "passively" but can check entries and even change them and therefore adjust themselves, depending on the input. The consequence can be that a

page, a section or a data record cannot be saved or released, if one or several input component(s) they contain have not been filled correctly. The cause is displayed at the input component, an incorrectly filled input component is also marked in color. This functionality is also called "Dynamic forms", the rules for which are created by the template developer (see Chapter 7.4 page 103).

Enabling dynamic forms:

- The **recognition of states with invalid content**. For example, not only will it be possible to check whether a date has been entered with the correct format, but also whether, depending on the requirement of the respective input component, it has to be in the future (e.g. future deadlines) or in the past (e.g. press releases).
- **Hiding individual input fields** under certain conditions. In this way, for example, display or editing by specific user groups can be prevented or the editing of individual fields is only allowed in specific stores and regardless of the permissions of a user.
- **Establishing logical relationships between individual fields**, which lead to automatic changing of a dependent field. In this way, for example, it is possible to use a checkbox to choose between two suppliers to affect a dependent combobox which, depending on the choice, displays specific products only.

The check is performed immediately during the editorial work. Incorrect inputs are uniformly visualized for all input components. The display clearly differs from the other layout of the workspace, in order to clearly highlight the incorrect inputs. The template developer can define how such invalid elements are to be handled, e.g. whether they can be released, whether or not they are taken into account in a generation. In this way, operating concepts of earlier versions of FirstSpirit can be modified so that, e.g. under Version 5.0, pages with invalid references can no longer be released, which was still possible in previous versions of FirstSpirit.

If groupings are used, incorrect inputs are also visualized via the tab sheets. Equally, if several languages are used, the program also visualizes the language in which invalid input exists:

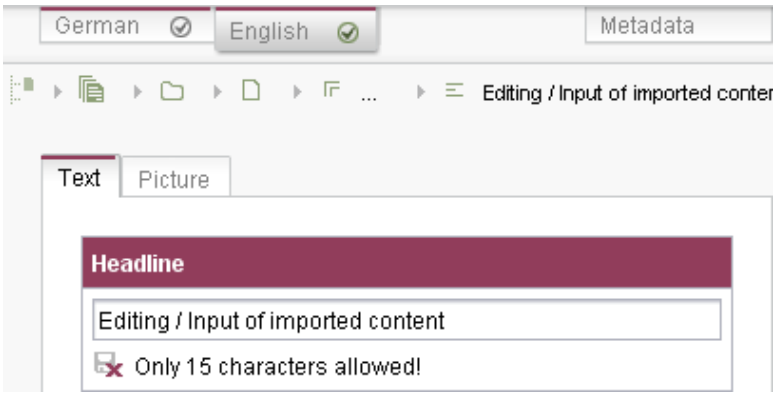


Figure 6-45: Visualization of invalid entries in the JavaClient

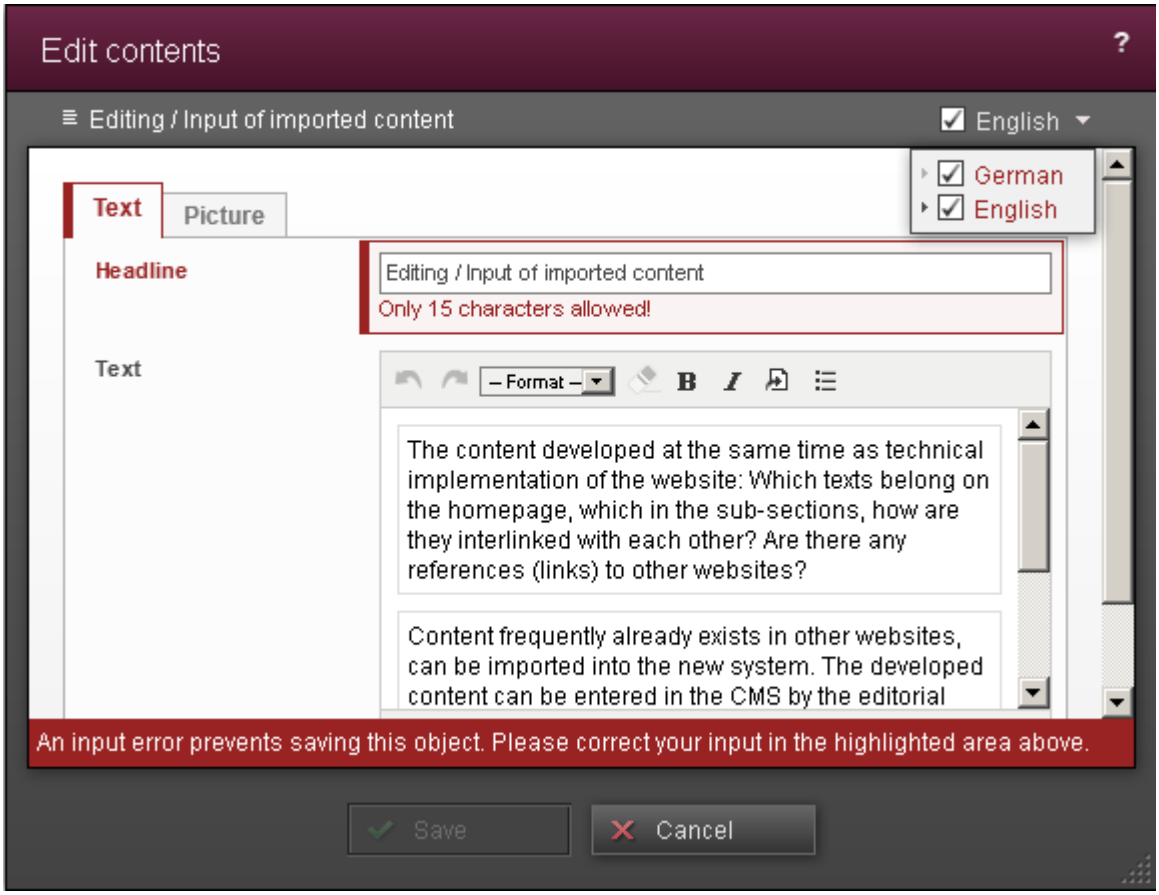


Figure 6-46: Visualization of invalid entries in the WebClient

Apart from color coding, the visualization also includes a textual, language-dependent reason why the state is invalid. If there are several incorrect entries, the respective number is also displayed, at the bottom end of each dialog. In the JavaClient the editor can click an entry to jump directly to the input component concerned, where they can then correct the input.

For further information, see FirstSpirit Manual for Editors (JavaClient) and FirstSpirit Online Documentation, "Dynamic Forms" chapter.

7 New/Modified functions for developers

7.1 WebClient 5.0

Project work is unavoidable when upgrading to WebClient 5.0 (see also Chapter 7.2 on page 99) because the required template changes ("migration") are not automatically performed in an upgrade. A key consideration is that some input components have been significantly expanded in their functionality in comparison with FirstSpirit 4 (e.g. FS_LIST), while a number of "old" components have disappeared (see Chapter 7.3.1 on page 101. In Version 5.0, the design of WebClient input components furthermore no longer pursued the concept of reproducing the JavaClient input components as closely as possible. Instead, our simplification philosophy was also applied on a component level. This for example concretely means that FS_LIST in the WebClient will **not** provide all the functions of FS_LIST in the JavaClient. This is entirely intentional and not a bug.!

- **Changing the object display:** So-called "snippets" can be used to change the manner in which
    - search results (global search and search in selection dialogs)
    - selections of references and templates
    - bookmarks
    - the project history
    - reports

  are displayed. See also Chapter 7.5.3 on page 107.
- **Using table templates:** It is now possible to specify for every table template if it is specifically meant to be usable in the WebClient (see *FirstSpirit Manual for editors (WebClient)*, Chapter "The FirstSpirit operating concept" etc.), by means of the "Use in WebClient" checkbox in the "Properties" tab. Before, all data sources were displayed in a tree view.
- **Mandatory fields** must be mapped in WebClient 5.0 using the "dynamic forms" also newly introduced in FirstSpirit 5.0 (see Chapter 7.4 on page 103).
- The **parameters** of some input components are no longer supported in WebClient 5.0 in contrast to the JavaClient and/or earlier versions of FirstSpirit, e.g.
    - CMS_INPUT_DATE: The parameter `allowInput`, which allows for manual data entry instead of date selection per dialog, is no longer supported in WebClient 5.0.
    - CMS_INPUT_DATE: The formats `w` (calendar week), `W` (week in month), `D` (day in year) and `F` (day of week in month) are not supported in the WebClient for the `format` parameter, which can be used to specify the manner in which time and

date information is shown in the input component. The display of linguistic values (`z`, `M`, `E`) depends on the client language specified in the start page.

- FS_DATASET: The `allowDelete` parameter has no effect in WebClient 5.0: this input component cannot be used to delete data records, but only the references to data records.

Further information of new and changed functions is also available in Chapter 5.1 on page 34 and in the WebClient Manual for Editors.

7.2 Template adjustment for Content Highlighting and Easy-Edit

In FirstSpirit Version 4.2, the "Content Highlighting" was introduced in the JavaClient and "Easy-Edit" in the WebClient, each of which enables closer linking between editorial maintenance and the output in the preview or on the generated page. Among other things, to this end, existing content is highlighted in the preview by a frame. In the WebClient there are also icons for editing section and data record content.

To use this functionality in FirstSpirit 5.0, the templates must be adjusted.

*However, it can still only ever be implemented for both Clients (Content Highlighting **and** Easy-Edit) simultaneously, as the same markup mechanisms are used for both in the templates.*

Therefore, in general, both aspects are implemented with only one markup. If WebClient 5.0 is used, however, it is necessary to check whether the required input component is supported by WebClient 5.0, as support is no longer available for several legacy components. In WebClient 5.0, mandatory fields must be mapped with the help of so-called "Dynamic Forms" (see Chapter 7.4 page 103).

In the JavaClient, if necessary Content Highlighting can be switched off using the "View" menu / "Content Highlighting Control" / "Disable". The template adjustment does not necessarily have to be made, as this involves an additional functionality, i.e. existing projects can continue to be used as usual, even without adjustments.

If Content Highlighting and / or Easy Edit are to be used in existing projects from Version 4.2.x, in which templates were already adjusted for the use of Easy Edit and / or Content Highlighting before Version 5.0, these adjustments from Version 4.2.x in the HTML output channel must be removed again. In addition, the format templates for Easy Edit and Content Highlighting (under

"Template Store" / "Format Templates" / "WebClient Format Templates (EasyEdit)" folder), supplied since Version 4.2, are no longer required and can also be removed. In Version 5.0 and higher, these format templates are also no longer supplied with FirstSpirit.

In FirstSpirit 5.0, content to be highlighted in the output is now marked up using the `editorId` parameter in the template sets.

Detailed documentation of the template adjustments required for use of Content Highlighting and Easy Edit in projects is provided in the FirstSpirit Online Documentation (ODFS), "Further Topics" / "Content Highlighting and EasyEdit" chapters.

7.3 Input components

7.3.1 Release of the new input components

With FirstSpirit Version 4.2, a fundamental revision and consolidation of FirstSpirit's input component model was started, which is completed in Version 5.0. The new input components with the prefix *FS_* introduced within this framework have taken on the functions of existing input components. The advantages of the new input components are that now, only one input component can now often be used for several use cases, which can therefore reduce the number of different page and section templates in the project. For example, if the editor is to be able to select pictures **and** PDF files, instead of two input components CMS_INPUT_PICTURE and CMS_INPUT_FILE, only FS_REFERENCE is required now.

The input component types replaced by the new "FS_" input components are **deprecated** with Version 5.0. This means that these "old" input components can still continue to be used in the initial released version of FirstSpirit 5.0, but they will no longer be further developed and only maintained to a limited extent (only limited bug fixes will be made) and will be removed completely in a later version. For WebClient, these input components are already no longer supported in Version 5.0.

It is therefore recommended that you change to the new input components!

This concerns the following input components:

Old:	Replacement:
CMS_INPUT_FILE	FS_REFERENCE
CMS_INPUT_PAGEREF	
CMS_INPUT_PICTURE	
CMS_INPUT_OBJECTCHOOSER	FS_DATASET
CMS_INPUT_CONTENTAREALIST	FS_LIST
CMS_INPUT_CONTENTLIST	
CMS_INPUT_LINKLIST	
CMS_INPUT_SECTIONLIST	
CMS_INPUT_TABLIST	

If uses of the input components deprecated in Version 5.0 are to be replaced by the new "FS_" input components in a project, this requires migrating the forms and output channels concerned. To find out how this is done, please refer to the ***FirstSpirit Release Notes for 4.2R4*, Chapter 5.1 ff.**

In addition, in FirstSpirit Version 4.2R4, the new FS_BUTTON input component was introduced. This will also not be officially released until Version 5.0.

7.3.2 FS_DATASET: New parameter

Use the parameter `mode` in the input component FS_DATASET to modify the selection and display mode. The mode `sheet` was added in In version 5.0. While all information of a data record are displayed by default (`mode="dialog"`) in text fields in the input component (see Figure 6-27) information can be displayed in a shortened manner by using `mode="sheet"`. By default, the name of the data source, the ID of the data record and the path are shown (see Figure 6-28).

If definitions were made for this input component on the tab "Snippets" of the respective table template on which the data record is basing (see Chapter 7.5.3 page 107), these are used for the

display of the information of the data record (see Figure 6-29).

7.3.3 CMS_INPUT_LINK: New parameter

In Version 5.0, the `mode` parameter has been introduced for the CMS_INPUT_LINK input component. This can be used to influence the display of the components. Possible values are:

`mode="dialog"`: This configuration is used to achieve the display of the input component used to date. To view stored information on the link (e.g. link target and link text), a dialog must be opened. Only the link text is displayed in the input component itself (see Figure 6-25). This is also the default setting, if `mode` is not set.

`mode="inline"`: With this configuration, all stored information on the link (e.g. link target and link text) is displayed directly in the form; it is no longer necessary to open an extra dialog (see Figure 6-26).

7.4 Dynamic forms (tab "Rules")

With FirstSpirit 5.0, functionalities have been implemented with which entries made in input components can be checked. In this way, stronger interaction with the user is also possible.

Input components now not only accept data "passively" but can check entries and even change them and therefore adjust themselves, depending on the input. The consequence can be that a page, a section or a data record cannot be saved or released, if one or several input component(s) they contain have not been filled correctly. The cause is displayed at the input component, an incorrectly filled input component is also marked in color. This functionality is also called "Dynamic forms", the rules for which are created by the template developer.

Enabling dynamic forms:

- The **recognition of states with invalid content**. For example, not only will it be possible to check whether a date has been entered with the correct format, but also whether, depending on the requirement of the respective input component, it has to be in the future (e.g. future deadlines) or in the past (e.g. press releases).
- **Hiding individual input fields** under certain conditions. In this way, for example, display or editing by specific user groups can be prevented or the editing of individual fields is only allowed in specific stores and regardless of the permissions of a user.
- **Establishing logical relationships between individual fields**, which lead to automatic changing of a dependent field. In this way, for example, it is possible to use a checkbox to choose between two suppliers to affect a dependent combobox which, depending on the

choice, displays specific products only.

The check is performed immediately during the editorial work. Incorrect inputs are uniformly visualized for all input components. The display clearly differs from the other layout of the workspace, in order to clearly highlight the incorrect inputs. The template developer can define how such invalid elements are to be handled, e.g. whether they can be released, whether or not they are taken into account in a generation. In this way, operating concepts of earlier versions of FirstSpirit can be modified so that, e.g. under Version 5.0, pages with invalid references can no longer be released, which was still possible in previous versions of FirstSpirit.

You can find a detailed documentation of the usable syntax in the FirstSpirit Online documentation, Chapter "Dynamic forms".

7.5 Search

With FirstSpirit Version 5.0 there are new search functionalities, which can be influenced by the template developer.

7.5.1 Global search

The global search, which is started using the Search window in the main view, (see Figure 6-1), is no longer displayed in a separate dialog, but instead directly in the left-hand area of the (see also Chapter 6.1.2 page 38).

The template developer can influence the display of the search results for

- pages, page references, sections, data records via the respective template ("Snippets" tab, see Chapter 7.5.3 page 107) and
- scripts via the "Comment" field.

7.5.2 Search syntax

Using the following search syntax you can not only search through the full text but only non-textual information, e.g. where are input components in the Page Store or Content Store with a specific identifier which are not filled. The search syntax allows also search for figures, dates and the use of relational operators.

Search area	Syntax	Search result
date specification	`yyyy-mm-dd hh:mm:ss +-zzzz`	Use the stated syntax to search for datum and / or time; you can also search only for parts of a date or time, e.g. `10` finds all date and time occurrences containing the figure 10 and full-text
metadata	fs.meta=0	nodes on which no metadata are defined
	fs.meta=1	nodes on which metadata are defined
release status	fs.releaseStatus=0	released nodes
	fs.releaseStatus=1	nodes which are not released
	fs.releaseStatus=3	nodes which are new or which have never been released yet
workflow	fs.workflow=x	nodes on which the workflow with the ID x has been started
images	fs.width=x	images with an original resolution of x pixel width
	fs.height=x	images with an original resolution of x pixel height

filled input components	`IDENTIFIER`=*	Input components in the Page store and Content store with the identifier `IDENTIFIER` with content
empty input components	`IDENTIFIER`=""	Input components in the Page store and Content store with the identifier `IDENTIFIER` with no content
Input components with specific textual content	`IDENTIFIER`=Text	Input components in the Page store and Content store with the identifier `IDENTIFIER` with the content `Text`
Input components with numeric content		Input components in the Page store and Content store with the identifier `IDENTIFIER` of the type CMS_INPUT_NUMBER containing
	`IDENTIFIER`=Figure	the indicated figure
	`IDENTIFIER`<Figure	a figure which is less than the indicated figure
	`IDENTIFIER`>Figure	a figure which is greater than the indicated figure
	`IDENTIFIER`<=Figure	a figure which is less than or equal to the indicated figure
	`IDENTIFIER`>=Figure	a figure which is greater than or equal to the indicated figure
	`IDENTIFIER`==Figure	exactly the indicated figure

7.5.3 Presentation of search hits (tab "Snippets")

For some template types, how search results are to be displayed based on those templates (e.g. pages, sections, data records) can be specified using the "Snippet" tab. The variable names for input components on the template are referenced for this.

The goal is not just to display search hits with object names (i.e., depending on the hit, the page name, section name, data record ID, etc.), but rather mirroring the respective object as accurately as possible using

- an image,
- a title and
- a text excerpt.

This way, the editor is able to receive a clear presentation of the content of the search hit in order to determine the most relevant hit more easily and to get to the object being searched for more quickly. This representation is used in both JavaClient and WebClient.

For detailed information about the usable syntax and its effects in the project see also FirstSpirit Online Documentation, Chapter "Snippets".

 *Definitions on the "Snippets" tab of meta data or project settings templates do not have any effect on the presentation of search hits.*

7.6 Code completion for forms

To provide template developers with greater support when they are creating forms, with Version 5.0 code completion has been introduced on the Form tab. With this code completion, all available FirstSpirit input components and all corresponding parameters with the available values can be displayed at the press of a key and inserted at the insertion point on the Form tab, e.g.

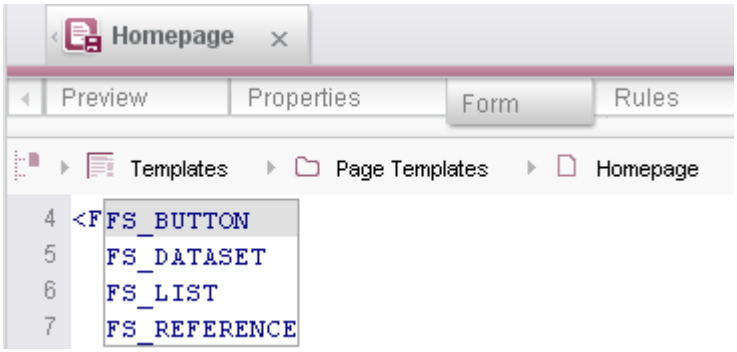


Figure 7-1: Auto-completion of the Form tab

For this, the insertion point must be positioned within a `<CMS_MODULE>` tag.

You can read about the tags and parameters for the input components, data and design elements with the respective values and their syntax and meaning in the FirstSpirit Online Documentation, "Template Development" / "Forms" chapter.

7.6.1 Adding the input component tags

In order to determine the input component tags (`FS_...` or `CMS_...`), an angle bracket (`<`) must be opened and the insertion point positioned behind it. The tags are then displayed in a list, if <Ctrl> and the space bar are pressed simultaneously. The required tag can then be copied onto the Form tab using the keyboard (Up or Down cursor key and <Enter>) or the mouse (double-click or click and <Enter>). The opening and closing tag and mandatory parameters (usually `name`) are inserted, e.g. for selection of `FS_BUTTON`:

```
<FS_BUTTON name=""></FS_BUTTON>
```

The insertion point is then located between the quotation marks of the `name` parameter.

The number of tags shown can be limited by entering the first letter(s) of the required input component behind the angle bracket, e.g. `<C` for the input components beginning with "CMS_" or `<F` for those beginning with "FS_".

Crossed-out entries in the list are obsolete and should not be used.

7.6.2 Adding tags, parameters and key terms

In order to be able to display and select the available tags, parameters and key terms of an input component, depending on the form syntax, the insertion point must be positioned as follows:

- **in opening tags:** To display parameters within an opening tag, a space must exist in front of the insertion point.
- **between the opening and closing tag:** To display tags between the opening and closing tag, an angle bracket must be opened in front of the insertion point (<).
- **within quotation marks:** In order to display values predefined by FirstSpirit ("key terms") of a parameter, the insertion point must be positioned within the quotation marks.

Only the tags, parameters and key terms available for the selected tag or parameter are ever displayed. Tags or parameters already used for the form, which can only be used once, are no longer displayed in the list.

If the required tags, parameters and key terms are already known, the first letter(s) can also be entered. The number of entries to be selected is then reduced with <Ctrl> + space bar or the entry is added directly. Where possible, mandatory parameters are also inserted directly.

Crossed-out entries in the list are obsolete and should not be used.

7.7 Optimized management of media (Exif data)

Images and files with different formats can be filed and managed in the FirstSpirit Media Store. With FirstSpirit 5.0, new infrastructure has been introduced for mime type recognition and an Exif library ("Exchangeable Image File Format"). The objective is to categorize media and to be able to more flexibly respond to files with the wrong file extension.

Depending on the configuration set by the project developer, this Exif data of imported pictures can be displayed on the "Metadata" tab of a picture and if applicable can also be changed, e.g. camera manufacturer and model, date and time taken, focal length, exposure time) (see Figure 6-30).

7.7.1 Form and output

To this end, a metadata template can be created or extended (see *FirstSpirit Online Documentation (ODFS)*, "Template development" / "Variables" / "Definition and output" / "in metadata" chapters). To do this, input components suitable for acquisition of the Exif data are defined on the Form tab, e.g.

Information	Input component
Text such as camera manufacturer and model	CMS_INPUT_TEXT
Numbers such as picture width, height, film sensitivity (ISO), aperture setting, GPS data	CMS_INPUT_NUMBER
Date / time e.g. date and time recorded	CMS_INPUT_DATE
Option selection e.g. picture orientation (top left/right, bottom left/right, etc.), flash settings (not triggered, automatic, switched on, etc.)	CMS_INPUT_COMBOBOX

For these input components, the identifier used (`name` parameter) must be the corresponding

tags as hexadecimal value, which are defined by the Exif standard (see http://www.exif.org/Exif2-2.PDF or information of the camera manufacturer). Each hexadecimal value must also be assigned the prefix `FS_` for processing by FirstSpirit, e.g.

Picture taking parameters	**Identifier (`name` parameter)**
Camera manufacturer	FS_0x010F
Camera model	FS_0x0110
Orientation	FS_0x0112
Horizontal resolution	FS_0x011A
Vertical resolution	FS_0x011B
Resolution unit	FS_0x0128
Shutter speed	FS_0x829A
Film sensitivity to ISO	FS_0x8827
Date and time taken	FS_0x9003
Aperture setting	FS_0x9202
Focal length of lens	FS_0x920A

Example code for the metadata form:

```
<CMS_INPUT_TEXT name="FS_0x010F" hFill="yes" singleLine="yes"
       useLanguages="no">
 <LANGINFOS>
   <LANGINFO lang="*" label="Camera manufacturer (FS_0x010F)"/>
   <LANGINFO lang="DE" label="Kamerahersteller (FS_0x010F)"/>
 </LANGINFOS>
</CMS_INPUT_TEXT>

<CMS_INPUT_TEXT name="FS_0x0110" hFill="yes" singleLine="yes"
       useLanguages="no">
 <LANGINFOS>
   <LANGINFO lang="*" label="Camera model (FS_0x0110)"/>
   <LANGINFO lang="DE" label="Kameramodell (FS_0x0110)"/>
 </LANGINFOS>
</CMS_INPUT_TEXT>

<CMS_INPUT_COMBOBOX name="FS_0x0112" hFill="yes" singleLine="yes"
```

```
        useLanguages="no">
<ENTRIES>
  <ENTRY value="0">
        <LANGINFOS>
        <LANGINFO lang="*" label="Unknown"/>
        <LANGINFO lang="DE" label="Unbekannt"/>
        </LANGINFOS>
  </ENTRY>
  <ENTRY value="1">
     <LANGINFOS>
        <LANGINFO lang="*" label="Top left"/>
        <LANGINFO lang="DE" label="Oben links"/>
        </LANGINFOS>
  </ENTRY>
  <ENTRY value="2">
        <LANGINFOS>
        <LANGINFO lang="*" label="Top right"/>
        <LANGINFO lang="DE" label="Oben rechts"/>
        </LANGINFOS>
  </ENTRY>
  <ENTRY value="3">
        <LANGINFOS>
        <LANGINFO lang="*" label="Bottom right"/>
        <LANGINFO lang="DE" label="Unten rechts"/>
        </LANGINFOS>
  </ENTRY>
  <ENTRY value="4">
        <LANGINFOS>
        <LANGINFO lang="*" label="Bottom left"/>
        <LANGINFO lang="DE" label="Unten links"/>
        </LANGINFOS>
  </ENTRY>
  <ENTRY value="5">
        <LANGINFOS>
        <LANGINFO lang="*" label="Left top"/>
        <LANGINFO lang="DE" label="Links oben"/>
     </LANGINFOS>
     </ENTRY>
  <ENTRY value="6">
     <LANGINFOS>
     <LANGINFO lang="*" label="Right top"/>
      <LANGINFO lang="DE" label="Rechts oben"/>
    </LANGINFOS>
  </ENTRY>
  <ENTRY value="7">
     <LANGINFOS>
     <LANGINFO lang="*" label="Right bottom"/>
        <LANGINFO lang="DE" label="Rechts unten"/>
     </LANGINFOS>
 </ENTRY>
     <ENTRY value="8">
     <LANGINFOS>
     <LANGINFO lang="*" label="Left bottom"/>
      <LANGINFO lang="DE" label="Links unten"/>
     </LANGINFOS>
```

```
      </ENTRY>
   </ENTRIES>
      <LANGINFOS>
      <LANGINFO lang="*" label="Orientation (FS_0x0112)"/>
     <LANGINFO lang="DE" label="Ausrichtung (FS_0x0112)"/>
    </LANGINFOS>
</CMS_INPUT_COMBOBOX>

<CMS_INPUT_NUMBER name="FS_0x011A" type="long" hFill="yes"
             singleLine="yes" useLanguages="no">
   <LANGINFOS>
      <LANGINFO lang="*" label="Horizontal resolution
             (FS_0x011A)"/>
      <LANGINFO lang="DE" label="Horizontale Auflösung
             (FS 0x011A)"/>
 </LANGINFOS>
</CMS_INPUT_NUMBER>

<CMS_INPUT_NUMBER name="FS_0x011B" type="long" hFill="yes"
             singleLine="yes" useLanguages="no">
   <LANGINFOS>
      <LANGINFO lang="*" label="Vertical resolution (FS_0x011B)"/>
    <LANGINFO lang="DE" label="Vertikale Auflösung
             (FS_0x011B)"/>
   </LANGINFOS>
</CMS_INPUT_NUMBER>

<CMS_INPUT_COMBOBOX name="FS_0x0128" hFill="yes" singleLine="yes"
          useLanguages="no">
   <ENTRIES>
      <ENTRY value="0">
      <LANGINFOS>
      <LANGINFO lang="*" label="Unknown"/>
       <LANGINFO lang="DE" label="Unbekannt"/>
      </LANGINFOS>
   </ENTRY>
      <ENTRY value="1">
      <LANGINFOS>
      <LANGINFO lang="*" label="No-unit"/>
       <LANGINFO lang="DE" label="Keine Einheit"/>
      </LANGINFOS>
      </ENTRY>
      <ENTRY value="2">
      <LANGINFOS>
      <LANGINFO lang="*" label="Inch"/>
       <LANGINFO lang="DE" label="Zoll"/>
      </LANGINFOS>
      </ENTRY>
      <ENTRY value="3">
      <LANGINFOS>
      <LANGINFO lang="*" label="Centimeter"/>
          <LANGINFO lang="DE" label="Zentimeter"/>
      </LANGINFOS>
      </ENTRY>
   </ENTRIES>
```

```
   <LANGINFOS>
          <LANGINFO lang="*" label="Resolution unit (FS_0x0128)"/>
      <LANGINFO lang="DE" label="Auflösungseinheit
          (FS_0x0128)"/>
      </LANGINFOS>
</CMS_INPUT_COMBOBOX>

<CMS_INPUT_DATE name="FS_0x0132" hFill="yes" singleLine="yes"
             useLanguages="no">
  <LANGINFOS>
      <LANGINFO lang="*" label="Change date and time (FS_0x0132)"
          format="yyyy:MM:dd HH:mm:ss"/>
      <LANGINFO lang="DE" label="Änderungsdatum und -zeit
             (FS_0x0132)" format="yyyy:MM:dd HH:mm:ss"/>
  </LANGINFOS>
</CMS_INPUT_DATE>

<CMS_INPUT_NUMBER name="FS_0x829A" type="double" hFill="yes"
             singleLine="yes" useLanguages="no">
  <LANGINFOS>
      <LANGINFO lang="*" label="Exposure time (FS_0x829A)"
             format="#.########################"/>
      <LANGINFO lang="DE" label="Verschlusszeit (FS_0x829A)"
             format="#.########################"/>
  </LANGINFOS>
</CMS_INPUT_NUMBER>

<CMS_INPUT_NUMBER name="FS_0x8827" type="double" hFill="yes"
             singleLine="yes" useLanguages="no">
  <LANGINFOS>
      <LANGINFO lang="*" label="ISO speed rating (FS_0x8827)"
             format="#.########################"/>
      <LANGINFO lang="DE" label="Filmempfindlichkeit nach ISO
             (FS_0x8827)" format="#.########################"/>
  </LANGINFOS>
</CMS_INPUT_NUMBER>

<CMS_INPUT_DATE name="FS_0x9003" hFill="yes" singleLine="yes"
             useLanguages="no">
  <LANGINFOS>
      <LANGINFO lang="*" label="Creation date and time
             (FS_0x9003)" format="yyyy:MM:dd HH:mm:ss"/>
      <LANGINFO lang="DE" label="Aufnahmedatum und -zeit
             (FS_0x9003)" format="yyyy:MM:dd HH:mm:ss"/>
  </LANGINFOS>
</CMS_INPUT_DATE>

<CMS_INPUT_NUMBER name="FS_0x9202" type="double" hFill="yes"
             singleLine="yes" useLanguages="no">
  <LANGINFOS>
      <LANGINFO lang="*" label="Aperture (FS_0x9202)"
             format="#.########################"/>
      <LANGINFO lang="DE" label="Blendeneinstellung (FS_0x9202)"
             format="#.########################"/>
  </LANGINFOS>
```

```
</CMS_INPUT_NUMBER>

<CMS_INPUT_NUMBER name="FS_0x920A" type="double" hFill="yes"
            singleLine="yes" useLanguages="no">
  <LANGINFOS>
     <LANGINFO lang="*" label="Lens focal length (FS_0x920A)"
            format="#.######################"/>
     <LANGINFO lang="DE" label="Linsenbrennweite (FS_0x920A)"
            format="#.######################"/>
  </LANGINFOS>
</CMS_INPUT_NUMBER>
```

The Exif syntax including an example is described under http://www.w3.org/2003/12/exif/.

For further information on API enhancements in the Media Store see also Chapter 7.2 page 116.

7.7.2 Conditional display

Ideally, the input components for Exif data should only be displayed on the metadata template if it concerns a picture with Exif data or if values are available for this input component.

To this end, on the "Rules" tab (see also Chapter 7.4 page 103), the metadata template can be checked using the `PROPERTY` tags in conjunction with the `EMPTY` property, to establish whether the respective input component is empty. This can be hidden using the `VISIBLE` property.

Example for the `FS_0x010F` field (camera manufacturer):

```
<!-- hide empty Exif field 0x010F -->
<ON_EVENT>
  <WITH>
     <NOT>
        <PROPERTY source="FS_0x010F" name="EMPTY"/>
     </NOT>
  </WITH>
  <DO>
     <PROPERTY source="FS_0x010F" name="VISIBLE"/>
  </DO>
</ON_EVENT>
```

7.1 Template syntax

Some methods which have been deprecated in FirstSpirit 4.x have been dropped in FirstSpirit 5.0:

- Functions in the header
  - `CMSFont`
  - `genericHTMLTable`
  - `genericMenuGroup`
  - `GenericNavigation`
  - `genericPageGroup`
  - `include`
  - `legacyContains`
  - `legacyIf`
- Functions in instructions
  - `legacyCompare(...)`

7.2 API extensions

The FirstSpirit API documentations describe the interfaces with FirstSpirit available in the templates and scripts to access a wide range of values, formats, etc.

7.2.1 FirstSpirit Access API

As FirstSpirit Version 5.0 is a major release, a number of methods set to "deprecated" in earlier versions of FirstSpirit have now been dropped. The overall policy pursued in FirstSpirit 5.0 was to keep these API changes as small as possible.

Methods can furthermore also be set to this status in version 5.0 itself. Information on the methods concerned and their replacements is always available in the FirstSpirit Access API. Although methods with this status can be used, doing so is not recommended because they will be dropped as from FirstSpirit Version 5.1.

The API extensions provided over and above this include the following, amongst others:

- Through the method `setPreviewImage(byte[] bytes)` in the interface `Picture` or `File` in the package `de.espirit.firstspirit.access.store.mediastore`, a **thumbnail** for images or files can be set up.

7.2.2 FirstSpirit Developer API

The FirstSpirit Developer API is stable within a minor version line, i.e. the methods available in Version 5.0 can change in the next minor version change (to 5.1). See also the note about the unreleased API in Chapter 10.3 page 173.

Among other things, the following API extensions were performed:

7.2.2.1 WebClient

With the API it is possible to integrate both extensions of the WebClient user interface and also administrative extensions.

The user interface can be extended in the following way with plug-ins:

- **Entries in the "Action" menu**
  They make possible the execution of actions such as, for example, display of dialogs, queries of user data and use of API methods. The Action operates in the context of page reference which is shown in the preview.
- **Buttons in the EasyEdit and InEdit frames**
  They make possible the execution of actions such as, for example, display of dialogs, queries of user data and use of API methods. The Action operates in the context of store elements, to which the button has been added.
- **Reports**
  They make possible the display of project data or data from other sources (e.g. web services) and the use of filter possibilities for restricting the displayed datasets. The data can be extracted from the result display by means of drag-and-drop to an FS_BUTTON instance (see the "FS_BUTTON" point further below) and thus processes further.
- **FS_BUTTON**
  They permit the referencing of Java classes as handlers for click and drag-and-drop actions. In cooperation with report plug-ins, the FS_BUTTON instances can be used for further processing of report data and taking over of template-based elements.
- **Site status and workflow displays**
  They offer the possibility to convert a logic tailored along the project for display of the site status "Released", "Processed" and "In workflow". This extension can also summarize FirstSpirit elements in groups which can move WebClient users together in workflows or channel them further in an already started workflow.

Administrative extensions:

- **Storage-synchronization**
  It influences the site management of WebClient, whereby, in the process of creation or moving of a site/site reference pair based on the position of the site reference in the structure management, a suitable folder for contents management of the site element storage is determined.
- **Java classes for workflow activities**
  They make possible the use of Java classes with full access to the FirstSpirit APIs in the execution of a workflow activity.

The WebClient extension as well as the WebClient-specific API functionalities are described in details in the online documentation for FirstSpirit, "WebClient 5.0" / "Plug-In Development" area.

7.2.2.2 Search

All project-based data structures (media, sites, templates, etc.) are managed by FirstSpirit in a content repository, whereby each FirstSpirit project possesses its own self-contained content repository. The `QueryAgent` interface provides methods for defining queries for these content repositories. Thereby both object names and also contents (e.g. of sites, datasets or media) are searched, in the case of media also texts from the "Description" field (full-text search without differentiating between upper and lower case letters). In addition, the queries take into consideration also contents from the FirstSpirit data source management.

For the definition of a query, an instance of the `QueryAgent` type is necessary. An instance of the `QueryAgent` type can be requested through the `SpecialistsBroker` help of the `requireSpecialist(QueryAgent.TYPE)` method.

For further information see also FirstSpirit Development Manual for Components 5.0, Chapter *"Das Interface QueryAgent (Suchanfragen definieren)".*

7.2.2.3 URL generation

Interfaces and a new reference implementation, which offer the possibility to integrate various path strategies for URL generation in FirstSpirit, were introduced with FirstSpirit 5.0.

Along with the standard functionality for improving the URL generation (see Chapter 8.5 page 125), these interfaces can be used to implement new client-specific path strategies and integrate them as a module in FirstSpirit.

The reference implementation AdvancedUrlFactory, which is included in the delivery scope of FirstSpirit 5.0 and serves as a template for the implementation of client-specific modules, reads

out the URLs from a persistence structure (UrlRegistry) or, respectively, stores there newly generated URLs. The generation of the URLs is delegated to the `UrlFactory` interface (in the package `de.espirit.firstspirit.generate`). The calculated value is then stored in the UrlRegistry.

With the new method `getStoredUrl` in the interface `de.espirit.firstspirit.access.store.ContentProducer`, the generated URLs can be read out in the corresponding product contents.

The interface `UrlFactory` replaces the interface `UrlCreator`; `UrlCreator` receives in FirstSpirit Version 5.0 the "deprecated" status.

For information about the Interface `UrlFactory` and more, see FirstSpirit Development Manual for Components 5.0, Chapter "Einflussnahme auf die URL-Erzeugung".

7.2.2.4 Application area of JavaClient

With the interface `de.espirit.firstspirit.agency.ServerInformationAgent`, starting from FirstSpirit 5.0, the **version of the FirstSpirit server** can be determined. With the methods

- `getMajor()`
- `getMinor()`
- `getBuild()`
- `getBranch()`

the individual data (major, minor, build and branch version) can be determined, and with

- `getFullVersionString()`

the complete version number, namely, for example, 5.0_BETA.31.51560 can be provided.

With the newly introduced method `getEngineType()` in the `BrowserApplication` interface, the application integration of the **engine type of the currently used browser** can be determined within the FirstSpirit framework. The version of the currently used browser can be returned as a string with the help of `BrowserApplication.getEngineVersion()`. See about that also the developer documentation *FirstSpirit AppCenter*, Chapter "Interface: Browser Application".

7.2.2.5 Miscellaneous

- New possibilities were created for the **control of JavaClient and WebClient** per API within the editing environment in order to create elements and open them for processing or to open forms for display and processing of data of a given element or data from another source.
  See about that, for example, the following packages and interfaces:
  - Package `de.espirit.firstspirit.ui.operations`
  - Interface `FormsAgent`
    (Package `de.espirit.firstspirit.agency`)
  - Interface `SelectStoreElementOperation`
    (Package `de.espirit.firstspirit.ui.gadgets.aspects.operations`)
  - With the `DeploymentUtil` class in the package `de.espirit.firstspirit.access.schedule`, it can be determined which project contents must be newly generated.
- For reading out and processing of **Exif data of media**, the following interfaces were created:
  - `ExifData`
  - `ExifTag`
  - `ExifTags`
  - `ExifUploadHook`
  - `ExifValue`
  - `ExifValueFormatter<T>`
  - `ExifValueFormatter.ComplexExifValueFormatter<S,C>`
  - (all in the package
    `de.espirit.firstspirit.server.mediamanagement.exif`)

8 New/modified functions for administrators

8.1 New and changed Java VM and wrapper parameters (`fs-wrapper.conf`)

The configuration file `fs-wrapper.conf` in the `conf` directory is responsible for starting and stopping the Java process. A number of parameters have now been newly added and/or changed in FirstSpirit 5.0. While these parameters will be directly in use in the case of new installation, naturally, new parameters must be transferred to the existing configuration if existing installations are merely **upgraded**. The same is applicable for the configuration file `fs-wrapper-slave.conf`

In doing so there is a need to ensure that the existing `fs-wrapper.conf` file of the FirstSpirit server to be updated is backed up and that only customer-specific adjustments of heapsize parameters `initmemory`, `maxmemory`, `Xmn` and `PermGen` are manually imported to the `fs-wrapper.conf` file of the newly installed FirstSpirit 5 server from the backed up file. More information on in-place upgrades is also provided in Chapter 4.2 page 29.

If manual extensions are provided in the `fs-wrapper.conf` file, the numbering of wrapper parameters is no longer required to be consecutive as from FirstSpirit Version 5.0 (Java Service Wrapper Version 3.3.6 - parameter `wrapper.ignore_sequence_gaps=true`). Extensions should be inserted after the standard parameters, if possible, for greater clarity.

The following parameters have been newly added in the standard configuration of Version 5.0:

- `wrapper.ignore_sequence_gaps=true`
- `wrapper.java.additional.auto_bits.solaris=true`
- `wrapper.java.additional.X=-Djava.io.tmpdir=work`
- `wrapper.java.additional.X=#-XX:+UseCompressedOops`
- `wrapper.java.additional.X=-XX:+NeverTenure`
- `wrapper.java.additional.X=-XX:InitialCodeCacheSize=128M`
- `wrapper.java.additional.X=-XX:ReservedCodeCacheSize=128M`

The following parameters have been changed in comparison with earlier versions of FirstSpirit:

- `wrapper.ping.timeout`: The standard value of this parameter has been changed from 300 to 0.
- `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`
These parameters can be used to activate the logging of garbage collector calls. This logging process is now standard as from FirstSpirit 5.0.

- `wrapper.timer_slow_threshold=3`: The standard value for this parameter has been changed from 15 to 3.

The following parameters have been dropped from the standard configuration in comparison with earlier versions of FirstSpirit:

- `wrapper.java.additional.X=*.jmxremote.*`
  (z. B. `wrapper.java.additional.X=-Dcom.sun.management.jmxremote`)
- `wrapper.java.additional.X=-Dsun.rmi.dgc.server.gcInterval=3600000`
- `wrapper.java.additional.X=-Dsun.rmi.dgc.client.gcInterval=3600000`

`wrapper.java.additional.X=-XX:+CMSIncrementalMode`

The `X` in `wrapper.java.additional.X=` above is always a placeholder for a unique number.

A complete description of Java wrapper parameters and further information is available at: http://wrapper.tanukisoftware.org/doc/english/properties.html.

8.2 New security mechanisms for the protection of generated project content

Generated project content is stored by the FirstSpirit Server in the default generation directories (`fs5staging`, `fs5preview` and `fs5webedit`) or in local project generation directories. Access to these generation directories is protected; this means that user authentication is required when this content is opened, provided this is configured. These security mechanisms were extended with FirstSpirit Version 4.2.466 and now affect all global and local project staging web applications.

In Version 4.2.466 and higher, the new security mechanisms are automatically adopted for all global FirstSpirit standard web applications when the FirstSpirit Server is updated.

Automatic updating of the standard web applications does not work if an external application server is used, which does not have read and write access to the FirstSpirit Server directory `opt/firstspirit5/web`. In this case, the global web applications – as with each FirstSpirit update – must be updated manually, in order to use the staging enhancement.

After updating the FirstSpirit Server, local project web applications must also be updated and

deployed manually, in order to adopt the staging extension.

The new security settings can also affect modules, e.g. the "SEARCH" module. Therefore, indexing of the generated project content by a search engine is no longer possible without authentication information. In this case the configuration of the "SEARCH" module must be adjusted and the necessary login information added (for further information see Module Documentation for *FirstSpirit SEARCH*).

8.3 Exclusive editing of data records in content sources

Unlike other Stores, pre-FirstSpirit Version 5.0, content source nodes were not locked when data records were edited. In this way, several editors could simultaneously create new data records, edit, etc. in a content source. Parallel working within a database view is therefore advantageous for multi-user operation, but can result in conflicts, if two editors try to change the same data record at the same time. In this case, the following information is displayed: "The record has been changed by another editor. Your changes could not be saved!" The user can then choose whether their changes are nonetheless to be saved or not (see *FirstSpirit Manual for Editors (JavaClient)*, "Data entry" chapter).

With Version 5.0, a data record can be placed in Edit mode, so that no other editor can make changes to this data record. (See also Chapter 6.1.8 page 60.)

This option is **not** active by default and can be enabled in the project properties for the relevant project, in the "Options" area, via the "Exclusive editing in content sources" checkbox:

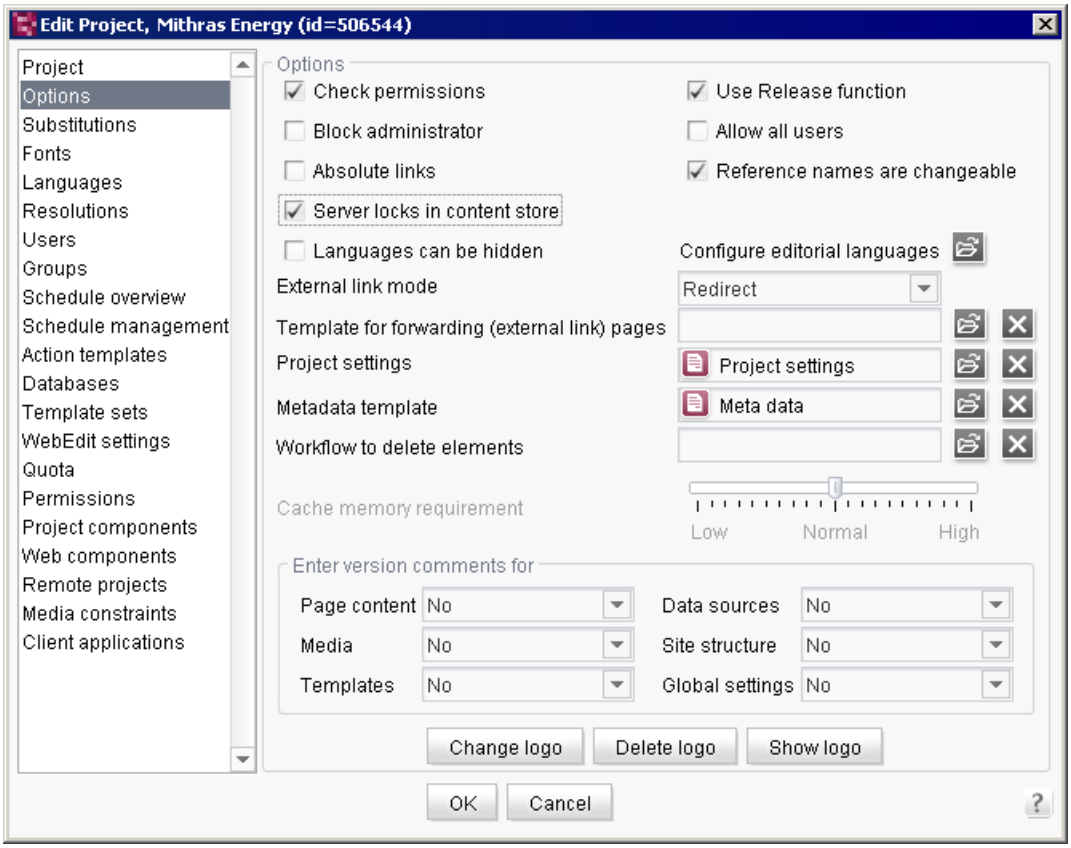


Figure 8-1: Project properties – "Exclusive editing in content sources" option

If this option is activated, it affects all content sources in the project concerned, not only in the JavaClient but also in the WebClient.

8.4 WebEdit settings

The WebClient 5.0 will not support projects that do not use **releases** (server and project configuration application, "Options", deactivated option "Use Release function").

New fields in the project settings:

- System folder
- Standard page template
- Workflow Group Provider
- Store Mapping
- Metadata

For more information about these options see FirstSpirit Manual for Administrators and FirstSpirit Online documentation.

8.5 Influencing URL generation (search engine optimization)

The naming of objects in FirstSpirit makes a strict distinction between display names (not unique, optionally maintainable mutlilingually, with Unicode support) and reference names (unique within the namespace, restricted to letters and numbers, i.e. no Unicode support). While the display names are relevant for the editing tasks and can be changed by the editor at any time, the reference names are normally only required by the template developer and for intrasystem actions and cannot be changed (or only with great effort).

This two-level naming has proven its worth in practical application, but means that reference names must be relied upon at certain junctures. URLs, for example, may not include Unicode characters as per specification. These structures are hence based on reference names in FirstSpirit and cannot be fully influenced by the editor for this reason.

As a consequence, FirstSpirit Version 5.0 now implements an option for exerting a greater influence on URLs than heretofore, e.g. for search engine optimization.

This functionality is deactivated by default in FirstSpirit 5 for compatibility reasons. The use of these functions may require adjustments in the project and/or organization that could also involve the live system (e.g. deployment scenarios, URL-based access protection or UTF-8 URLs). Detailed implementation planning is hence mandatory for this.

8.5.1 General information on the generation process in FirstSpirit

By default, the results of generation processes are filed in the FirstSpirit directory structure at

```
firstspirit5/web/fs5staging/[project ID]/[schedule ID]/...
```

The placeholder `..` will be used for this path in the directory specifications to follow.

Generation processes are principally based on the Site Store of a project or the page references, to put it more exactly. In the course of the generation process, these are converted – depending on the template set (also "channel") in the project – into files in above directory, e.g. HTML pages or PDF documents, in accordance with the specifications in the respective template files. The folder structure (menu levels) is also mapped in the generation process, namely as a folder in the directory structure.

If media are used in the generated pages, these will also be included in the generation process and filed in directories.

If the release option is used in a project, only nodes that have been released at least once will be included in the generation process as a matter of principle. In this case only released contents will be generated by default. This creates the need to ensure before the generation process that all changed nodes whose new contents are to be shown in the generated page have been released.

A standard schedule created for each project allows the implementation of a full generation process (schedule "generate full") or a partial generation process (schedule "generate partly"). Like any schedule, these schedules can be controlled by means of the project properties ("schedule management"). The schedules can also be started interactively from the JavaClient (menu "Project" / "Generate project" and/or "Generate partial project") if the relevant rights to do so are provided. Individual generation schedules can furthermore also be created in the schedule entry planning. These can also be started via the server and project configuration application and – given the required configuration – in the JavaClient using the menu option "Execute schedule entry".

These generation schedules allow specification of the manner in which paths are to be generated for FirstSpirit elements. The URLs the FirstSpirit contents can be accessed with later on are in turn based on these paths. FirstSpirit offers the following path generation options by default:

- Default URLs / Default URLs (SEO) (see Chapter 8.5.3 on page 129)
- Advanced URLs (see Chapter 8.5.4 on page 135)
- Infix URLs / Infix URLs (SEO)
- Multiview URLs / Multiview URLs (SEO)

Customer-specific or project-specific path generation solutions can also be implemented, however, depending on the needs and requirements. See the *FirstSpirit Documentation for Module Developers* for more information on this.

For better understanding, the new "Advanced URLs" path generation mode in FirstSpirit 5.0 is here being compared with the already familiar "Default URLs" mode (Chapter 8.5.3 on page 129 and/or 8.5.4 on page 135).

8.5.2 Storing and resetting URLs

URLs are a listing and ranking criterion for search engines and their correct design is hence an important part of search engine optimization (SEO). Because later changes in URLs that have already been indexed by search engines may affect the search engine page ranking, and pages with changed URLs may hence be rendered inaccessible for a period of time, URLs should remain in existence for as long as possible even if the website structure or page name have changed.

FirstSpirit 5.0 therefore introduces the option of storing generated URLs. This means that the URLs created in a generation process will remain valid even if the values of relevance for the URL generation process (depending on the path generation method selected, for example changed display or reference names, relocated pages or media in a tree, changes in data records) should change. These changes will only be reflected in the URL once the generated URLs have been manually reset. To store URLs, the "(SEO)" variant of the desired method must be selected in the generation settings. The variants without an added "(SEO)" will behave as before: the URLs will not be stored.

If the "(SEO)" variants are selected, settings in the tab "URL settings" / "Short URLs" (see Chapter 6.1.15.2 page 83) will also be taken into account in the generation process.

If changes in display names or the node structure are to be reflected in saved URLs, the menu option "Reset stored URLs" must be selected in the context menu under "Extras" at the node concerned in the JavaClient before the generation process. This is available for the following objects:

- Media
- Media Store folders
- Page references, and
- Menu levels

If media, page references or menu levels have been moved around within the tree structure or renamed and these changes are to be reflected in the URL after a generation process has already taken place, these changes will therefore only become effective once the objects in question have been released and undergone the function "Reset stored URLs". For changes in data record contents to be reflected in the URL, the function needs to be applied to the page reference where the attendant content projection is provided. All content changes that have no

impact on the URL will meanwhile of course be included in the next generation process following their release as usual.

The function also affects elements subordinated to the element in the tree structure where the function "Reset stored URLs" is invoked. This for example means that all the stored URLs of a tree section can be reset by invoking the function for the uppermost menu level, starting from which the URLs are to be reset. In this case the stored URLs of all the page references and menu levels subordinated to this menu level will be cleared.

The successful implementation of the function is followed by the message below:

Figure 8-2: Stored URLs have been deleted

Project administrator rights are required to implement this function.

If the "URL Settings" node in the "Global Content Area" is in edit mode, the context menu option "Reset stored URLs" will be deactivated for all nodes. The saved URL for the desired node can only be reset once the edit mode for the "URL Settings" is exited.

If and which URLs are stored in a node can be viewed by selecting the context menu option "Extras" / "Display properties" or by pressing Alt + P at the respective node.

8.5.3 Standard URL generation

This Chapter describes the results of a generation process in the "Default URLs" mode.

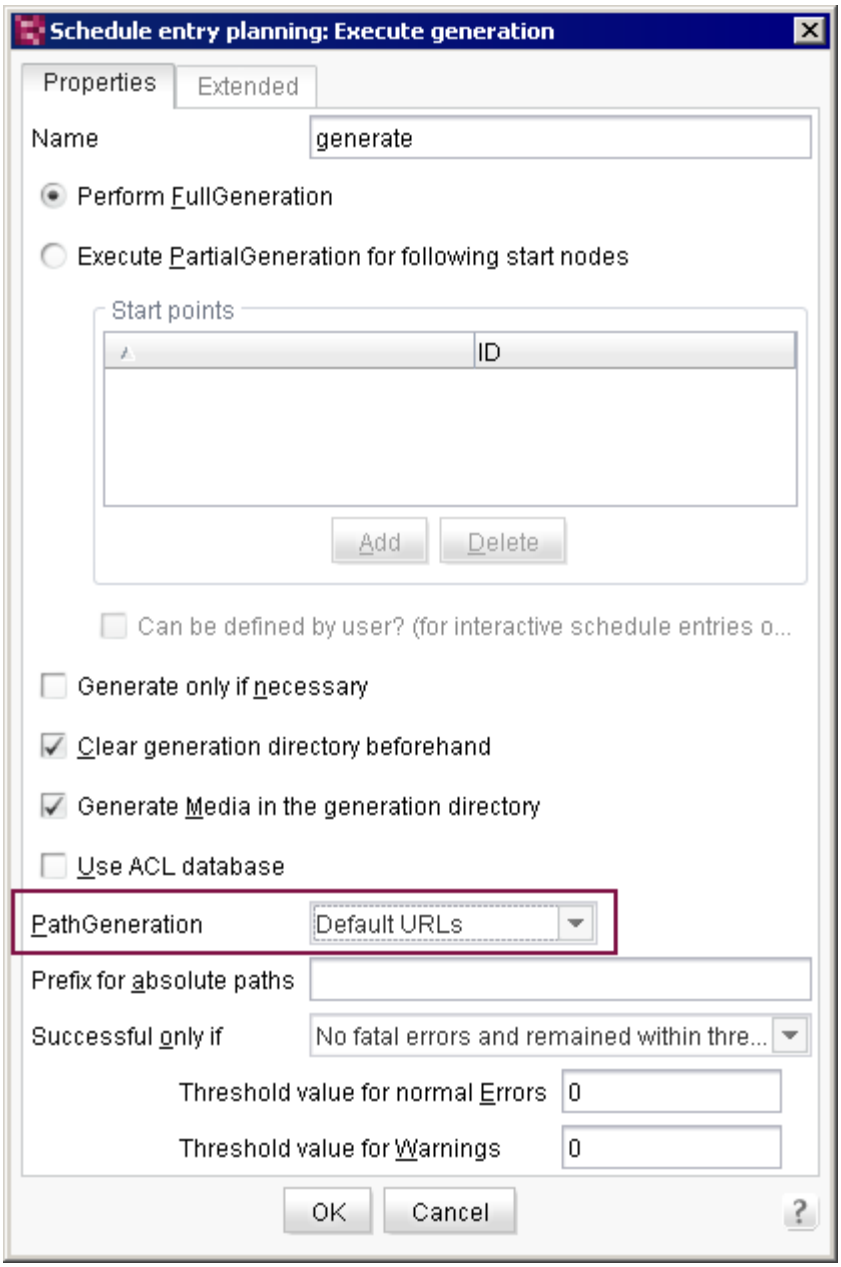


Figure 8-3: Generation – "Default URLs" path generation

Each language and each template set is provided with a directory positioned at the top level, i.e.

```
../de/...
```

and

```
../de_1/...
```

for two presentation channels (e.g. HTML and PDF) in the language DE and

```
../en/...
```

and

```
../en_1/...
```

for two presentation channels (e.g. HTML and PDF) in the language EN.

8.5.3.1 Menu levels and page references

Within the language directories, the Site Store objects to be included are filed hierarchically. This means that every released **page reference** is generated including the folder path (**menu levels**). Menu levels without page reference are not included in the generation process as long as they do not contain any other menu levels with released page references. Their name is derived from the file name assigned to the page reference in the Site Store, and from the reference name in the case of folders. As both file names and reference names are language-independent, URLs cannot be generated in a completely multilingual fashion, e.g.

```
../de/startpage/mithras_home.html
```

and

```
../en/startpage/mithras_home.html
```

Owing to the creation of menu levels per drag and drop from the Page Store, reference and file names in the Site Store are frequently formed from the reference names of folders and pages in the Page Store.

8.5.3.2 Page groups

If **page groups** are used, the page references contained are filed in a folder below the respective language folder. The reference name is relied upon in the process, i.e.

```
../de/pagegroup/pressemitteilung_1.html

../de/pagegroup/pressemitteilung_2.html

../de/pagegroup/pressemitteilung_3.html
```

and

```
../en/pagegroup/pressemitteilung_1.html

../en/pagegroup/pressemitteilung_2.html

../en/pagegroup/pressemitteilung_3.html
```

8.5.3.3 Data records

If **data records** are output across several pages by way of a content projection ("Content" tab on menu levels), a number will be added to the file name of the page reference where the content projection is output. If one data record is output per page, this will be the ID of the data record, e.g.

```
../de/press/pressreleases/press_releases_details_128.html
```

and/or

```
../en/press/pressreleases/press_releases_details_128.html
```

If several data records are output in one page, consecutive numbers are added, e.g.

```
../de/press/pressreleases/pressreleases.html

../de/press/pressreleases/pressreleases_1.html

../de/press/pressreleases/pressreleases_2.html
```

and

```
../en/press/pressreleases/pressreleases.html

../en/press/pressreleases/pressreleases_1.html
```

```
../en/press/pressreleases/pressreleases_2.html
```

8.5.3.4 Media

If media are included in the generation process (activated option "Generate media in generation directory" in the generation schedule) they will also be filed in a directory of their own at the top level:

```
../media/...
```

The Media Store media to be included will be filed in this directory hierarchically. Media folders will also be included in the directory structure, e.g.

```
../media/products/powerinverter/control-panel.jpg
```

The file name is used for media and the reference name for media folders. Images are furthermore filed in the used resolutions, with the name of the resolution appended to the file name of the image, e.g.

```
../media/products/powerinverter/control-panel_Produktteaser.jpg
```

Language-dependent media will be filed in separate language directories, e.g.

```
../media/de/products/downloaddokumente/produktuebersicht.doc
```

and

```
../media/en/products/downloaddokumente/produktuebersicht.doc
```

Inside these language directories, the structure of the Media Store is included again, i.e. the folder structure down to the media in question is also being mapped.

The file names and reference names are also language-independent in the Media Store so that URLs cannot be generated in a completely multilingual fashion.

8.5.3.5 Examples

The following example is based on the "Mithras Energy" demo project. For a comparison with example URLs in the mode "Advanced URLs", see Chapter 8.5.4.5 on page 142.

Example URLs for page references	
`../de/startpage/mithras_home.html`	Page reference DE, template set HTML
`../en/startpage/mithras_home.html`	Page reference EN, template set HTML
`../de/aboutus/company/company_1.html`	Page reference in subdirectory DE, template set HTML
`../en/aboutus/company/company_1.html`	Page reference in subdirectory EN, template set HTML
`../de_1/startpage/mithras_home.pdf`	Page reference DE, template set PDF
`../en_1/startpage/mithras_home.pdf`	Page reference EN, template set PDF
Example URLs for page groups	

`../de/press/pressreleases/pressrelease_1.html`
`../de/press/pressreleases/pressrelease_2.html`
`../de/press/pressreleases/pressrelease_3.html`
	Page group with 3 pages DE

`../en/press/pressreleases/pressrelease_1.html`
`../en/press/pressreleases/pressrelease_2.html`
`../en/press/pressreleases/pressrelease_3.html`
	Page group with 3 pages EN

Example URLs for multipages / content projection	
`../de/press/pressreleases/pressreleasesdetails_130.html`	Page with 1 data record (ID 130) DE
`../en/press/pressreleases/pressreleasesdetails_130.html`	Page with 1 data record (ID 130) EN

`../de/press/pressreleases/pressreleases.html`
`../de/press/pressreleases/pressreleases_1.html`
`../de/press/pressreleases/pressreleases_2.html`
	Pages with several data records DE

../en/press/pressreleases/pressreleases.html ../en/press/pressreleases/pressreleases_1.html ../en/press/pressreleases/pressreleases_2.html	Pages with several data records EN
Example URLs for media	
../media/products/powerinverter/control-panel.jpg	Language-independent image
../media/ products/powerinverter/control-panel_ Produktteaser.jpg	Resolution of a language-independent image
../media/layout/css/print.css	Language-independent file
../media/de/layout/logo.png	Language-dependent image DE
../media/en/layout/logo.png	Language-dependent image EN
../media/de/company/building.png	Language-dependent image DE
../media/en/company/building.png	Language-dependent image EN
../media/de/products/downloaddokumente/produktuebersicht .doc	Language-dependent file DE
../media/en/products/downloaddokumente/produktuebersicht .doc	Language-dependent file EN

8.5.4 Individual URL generation (e.g. "Advanced URLs")

FirstSpirit Version 5.0 comes with a newly designed API interface allowing the creation of own "URL generators" (e.g. in the form of modules or scripts) that enable the generation of exactly customized URLs. This can for example serve to support the search engine optimization (SEO), e.g. by reliance on "expressive" URLs that are more easily understood by website visitors and may get a better ranking from search engines. In addition, URLs can now be generated in a completely multilingual fashion. The planning of the URL structure should be started as early as possible in any case: later changes in the URLs of pages which have already been indexed can have a negative impact on their ranking (at least in the short term).

FirstSpirit 5.0's basic package includes a new URL generator, the "Advanced URL Creator". This provides merely an **example** for an URL creation strategy and lays NO claim to being able to map all SEO strategies in their entirety (which would be impossible anyway owing to the principles involved). Instead, it provides a basis allowing the creation of much more flexible URLs than in FirstSpirit Version 4. Java programming is required for customizing URLs to the respective requirements.

 *A complex URL generation strategy makes it impossible to predict possible conflicts (identical file names) arising from user activities (e.g. relocation of nodes within the tree structure). These conflicts only become detectable at a very late stage (e.g. when the file is generated or even as late as upon publication). This means that an inept implementation and/or inept selection of display names can create problems which would not arise in the standard URL creation process (see Chapter 8.5.3 page 129. These problems are detected and overcome by familiar disambiguation mechanisms (appending a consecutive number).*

 *Changes in the URL creator affect the downstream publication processes and live system configuration, which may need to be adapted to the modified URL creator.*

The "Advanced URL Creator" can be activated in generation schedules by way of the "PathGeneration" combobox (the default setting is "Default URLs", Figure 8-3, for greater compatibility with earlier versions of FirstSpirit):

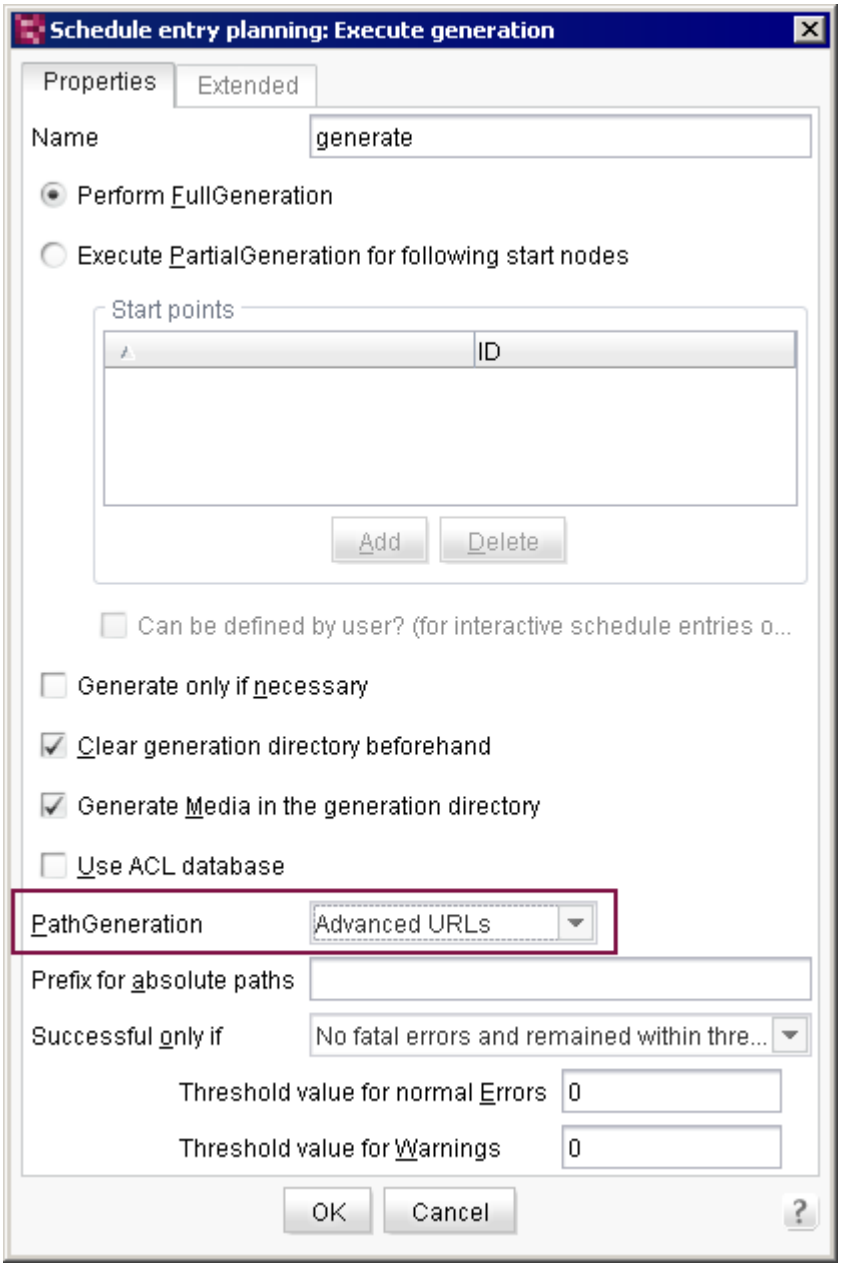


Figure 8-4: New path generation method – Advanced URLs

While the URLs created in the standard URL generation (see Chapter 8.5.3 page 129) are based on file and reference names, the URLs generated by the Advanced URL Creator are based on the display names of the FirstSpirit objects. This means that all URLs are created in UTF-8, including blanks and special characters. This can lead to problems in Windows because the file system is case-insensitive. Leading and final blanks are removed when the URLs are generated, however, and the following characters will be replaced by a minus (-):

```
\ / : * ? " < > | #
```

All URLs in a project can be realized multilingually throughout. The division into language directories and template set directories has been dropped. Any URLs/paths that are not unique are automatically rendered so in the generation process and hence differentiated from one another (by appending numbers).

This URL creator is therefore primarily suitable for projects where all the respective display names in the Site Store and Media Store are also multilingual throughout in all the languages to be generated. If no display name is provided in a language, the display name of the master language is used or, if no display name is provided in the master language either, the reference name is used.

For URLs with umlauts, Tomcat webservers used in the Advanced URLs mode should be version 6.0.29 or higher to ensure correct handling.

8.5.4.1 Menu levels and page references

Every **page reference** that is earmarked for inclusion is generated including the folder path (**menu levels**). Menu levels without page reference are not included in the generation process as long as they do not contain any other menu levels with released page references. As the display names are relied upon by both the page references and menu levels, all URLs can be created in a completely multilingual manner.

```
../de/startpage/mithras_home.html
```

and

```
../en/startpage/mithras_home.html
```

in standard URL creation

become

```
../Startseite/index.html
```

and

```
../Startpage/index.html
```

in the Advanced mode

These pages can also be accesses by means of

```
../Startseite/
```

or

```
../Startpage/
```

If the editor has not assigned a display name to a menu level in one of the languages, the directory of the master language ("Master language fallback") and/or a directory with the reference name ("Reference name fallback") is used. If this should lead to the existence of several `index.html` files in a directory, a number will be appended to differentiate them ("Disambiguation fallback"), e.g.

```
../Startseite/index.html

../Startseite/index.1.html
```

This is also the case when folders in various languages have the same display name.

A script which precedes the generation process (schedule management in the project attributes) can for example ensure the creation of files that are based on the display name of the page reference, instead of the "index.*" files:

```
context.setProperty("#urlCreatorSettings",
Collections.singletonMap("usewelcomefilenames", "false"));
```

8.5.4.2 Page groups

If page groups are used, the start page will be assigned the file name index.html by default. The file names of other pages in the page group will be generated from the display name of the page. If no display name is provided, the reference name will be used. The disambiguation of file names in folders follows the same procedure as for page references (see Chapter 8.5.4.1 page 137).

```
../de/pagegroup/pressemitteilung_1.html

../de/pagegroup/pressemitteilung_2.html

../de/pagegroup/pressemitteilung_3.html
```

and

```
../en/pagegroup/pressemitteilung_1.html

../en/pagegroup/pressemitteilung_2.html

../en/pagegroup/pressemitteilung_3.html
```

in the Standard URL mode

would hence become

```
../pagegroup/index.html

../pagegroup/Pressemitteilung 2.html

../pagegroup/Pressemitteilung 3.html
```

and

```
../pagegroup/index.html

../pagegroup/Press Release 2.html

../pagegroup/Press Release 3.html
```

in the Advanced URL mode

A script (schedule management in the project attributes) that precedes the generation process

```
   context.setProperty("#urlCreatorSettings",
Collections.singletonMap("usewelcomefilenames", "false"));
```

can also in this case ensure the creation of files that are based on the display name of the start page reference, instead of the "index.*" files.

8.5.4.3 Data records

If **data records** are output across several pages by way of a content projection ("Content" tab on menu levels), a number is added to the file name of the page reference where the content projection is output. If one data record is output per page, this will be the ID of the data record, e.g.

```
.../Presse/Pressemitteilungen/Pressemitteilungen Detailseite_128.html
```

and/or

```
.../Press/Press Releases/Press Releases Details_128.html
```

The entire folder path is also generated here. Similarly to the URL creation for page references

(see Chapter 8.5.4.1 page 137), the page references and folder path here are also based on the display names.

If a column of the data source is selected in the "Variable for sitemap text" field in a content projection (Site Store, page reference with "Content" tab), the name will be derived from the text in this column in a language-dependent fashion. This way "expressive URLs" can also be created for multi-pages, e.g.

```
../Presse/Pressemitteilungen/Neuer Geschäftsführer bei Mithras Energy.html
```

or

```
../Press/Press Releases/New Director of Mithras Energy.html
```

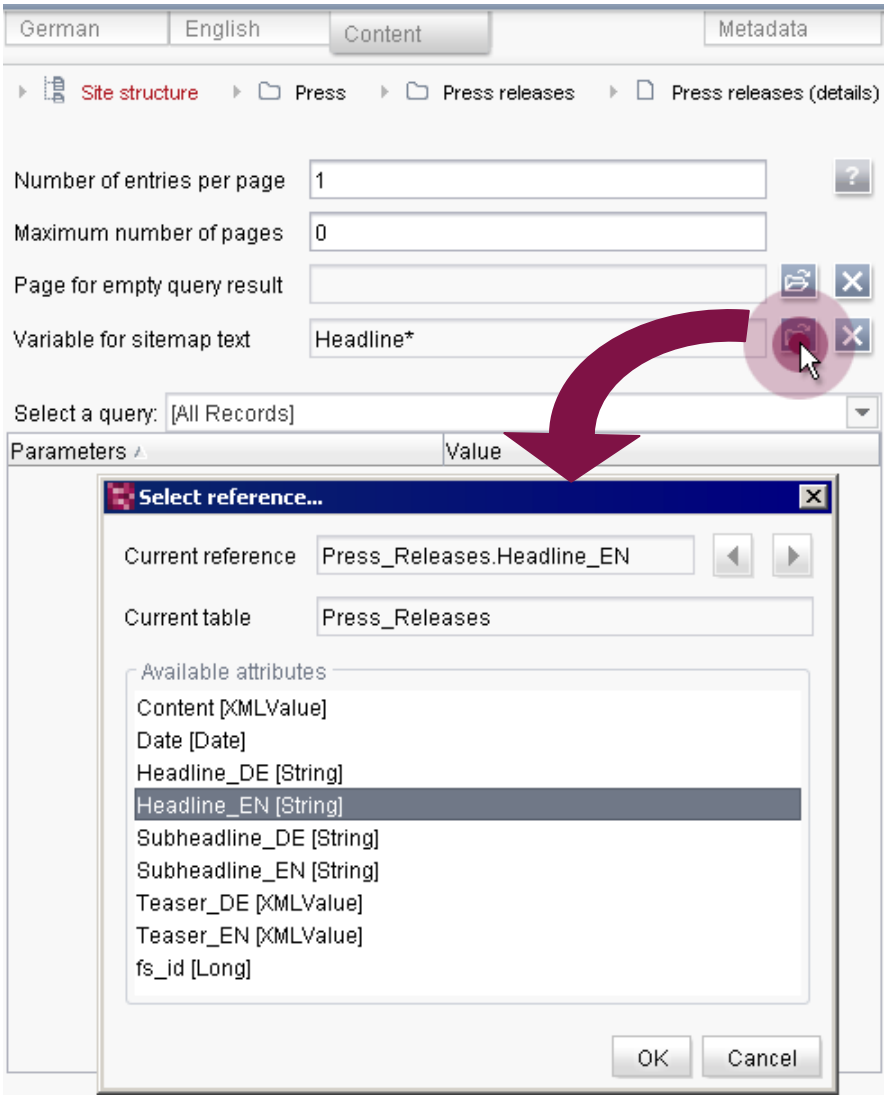


Figure 8-5: URL management for multi-pages

If the field selected in "Variable for sitemap text" is empty in one or several languages, the URL is once again formed by appending the data record ID by default. If several data records are output on one page, consecutive numbers will be appended, e.g.

```
../Presse/Pressemitteilungen/Pressemitteilungen Übersicht.html

../Presse/Pressemitteilungen/Pressemitteilungen Übersicht_1.html

../Presse/Pressemitteilungen/Pressemitteilungen Übersicht_2.html
```

or

```
../en/Press/Press Releases/Press Releases Overview.html

../en/Press/Press Releases/Press Releases Overview_1.html

../en/Press/Press Releases/Press Releases Overview_2.html
```

8.5.4.4 Media

In contrast to the standard URL generation, no superordinated media directory is created in Advanced URL generation. The Media Store **media** to be included are filed hierarchically in folders, while folders and/or media from the root directory of the Media Store are filed in the top level of the generation directory. The display names are relied upon for both media and media folders, meaning that all URLs can be created multilingually throughout. As a result, Site Store files (HTML pages, PDF documents, etc.) and media might be filed in the same directory if the display names of the folders are identical in the Site Store and Media Store.

These items in standard URL generation

```
1) ../media/products/powerinverter/control-panel.jpg

2) ../media/products/powerinverter/control-panel_Produktteaser.jpg

3) ../media/de/products/downloaddokumente/produktuebersicht.doc

4) ../media/en/products/downloaddokumente/produktuebersicht.doc
```

turn into this in Advanced URL generation

```
1) ../Produkte/Wechselrichter/Instrumententafel.jpg

2) ../Produkte/Wechselrichter/Instrumententafel_Produktteaser.jpg

3) ../Produkte/Word Download Dokumente/Produkt Übersicht.doc
```

```
4) ../Products/Word downloads/Product overview.doc
```

In the case of language-dependent media (a different medium is used for every language provided in the project if the reference is the same, examples 3) and 4)), the display name in the respective language and/or the reference name will be used. In the case of language-independent media (the same medium is used for all languages, examples 1) and 2)), the display name in the master language is used. If none is assigned, the reference name will be used.

If the editor has not assigned a display name to a medium in one of the languages, the display name in the master language ("Master language fallback") and/or the reference name ("Reference name fallback") will be used. If this should lead to the creation of several identically named files in a directory, a number will be appended to differentiate them ("Disambiguation fallback"), e.g.

```
../logo.png

../logo.1.png
```

URLs can also be manually assigned to media folders by means of the API, e.g. "media" for the media root node.

8.5.4.5 Examples

The following example is based on the "Mithras Energy" demo project. For a comparison with example URLs in the mode "Default URLs", see Chapter 8.5.3.5 page 133.

URL examples for menu levels / page references	
`../Startseite/(index.html)`	Page reference DE, template set HTML
`../Startpage/(index.html)`	Page reference EN, template set HTML
`../Über uns/Unternehmen/(index.html)`	Page reference in subdirectory DE, template set HTML
`../About us/Company/(index.html)`	Page reference in Subdirectory EN, template set HTML
`../Startseite/index.html`	Page reference DE, template set HTML (Reference name fallback)

../Startseite/index.1.html	Page reference EN, template set HTML (Master language fallback DE and disambiguation)
../Startseite/index.pdf	Page reference DE, template set PDF
../Startpage/index.pdf	Page reference EN, template set PDF
URL examples for page groups	
../Presse/Pressemitteilungen/index.html ../Presse/Pressemitteilungen/Pressemitteilung 2.html ../Presse/Pressemitteilungen/Pressemitteilung 3.html	Page group with 3 pages DE
../Press/Press Releases/index.html ../Press/Press Releases/Press Release 2.html ../Press/Press Releases/Press Release 3.html	Page group with 3 pages EN
URL examples for multi-pages / content projections	
../Presse/Pressemitteilungen/Neuer Geschäftsführer bei Mithras Energy.html	Page with 1 data record (ID 130) DE
../Press/Press Releases/New Director of Mithras Energy.html	Page with 1 data record (ID 130) EN
../Presse/Pressemitteilungen/Pressemitteilungen Übersicht.html ../Presse/Pressemitteilungen/Pressemitteilungen Übersicht_1.html ../Presse/Pressemitteilungen/Pressemitteilungen Übersicht_2.html	Pages with several data records DE
../Press/Press Releases/Press Releases Overview.html ../Press/Press Releases/Press Releases Overview_1.html ../Press/Press Releases/Press Releases Overview_2.html	Pages with several data records EN
URL examples for media	
../Produkte/Wechselrichter/Instrumententafel.jpg	Language-independent image DE and EN
../Produkte/Wechselrichter/Instrumententafel_Produkteaser.jpg	Resolution of a language-independent image DE and EN

`../Layout/CSS/print.css`	Language-independent file DE and EN (Reference name fallback)
`../Layout/logo.png`	Language-dependent image DE (Reference name fallback)
`../Layout/logo.1.png`	Language-dependent image EN (Reference name fallback and disambiguation)
`../Unternehmen/Gebäude.png`	Language-dependent image DE
`../Company/Building.png`	Language-dependent image EN
`../Produkte/Word Download Dokumente/Produkt Übersicht.doc`	Language-dependent file DE
`../Products/Word downloads/Product overview.doc`	Language-dependent file EN

The paths generated by the Advanced URL Creator can be manually adjusted in the JavaClient (not for media, however). See also Chapter 6.1.15 page 75 for more information on this.

8.6 Automatic archiving log files of the Java Garbage Collector

The log file of the garbage collector of the JAVA VM on the FirstSpirit 5.0 server side will automatically be archived when a special size is reached, and the N-oldest archive file deleted at the same time.

The name of the log file is automatically read from the configuration line `wrapper.java.additional.X=-Xloggc:log/fs-gc.log` or `wrapper.java.additional.X=-Xverbosegclog` for IBM JDK contained in the file `fs-wrapper.conf.`

Once the defined file-size is reached (5 MB), the current log file will be both renamed by an addition that contains the date of the first entry and compressed, which is reflected in the additional extension `.gz` (e.g. `fs-gc.20120227_153639.log.gz`).

The following parameters of the fs-wrapper.conf file must be activated (and not commented out by #) to do this. This configuration is active for newly installed FirstSpirit 5.0 servers):

```
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=` above is always a placeholder for a unique number.

Archived files can be deleted or moved to `firstspirit5/backup` by using the server schedule "Clean up logs”.

8.7 Server update

With FirstSpirit 5.0, FirstSpirit Servers can be updated to new software versions via FirstSpirit Server Monitoring. This is also partly possible fully automatically.

e-Spirit provides software updates via

- **manual update** (see Chapter 8.7.1 page 147)
  The file/s which is/are necessary for the update is/are downloaded and the update will then be carried out via the FirstSpirit Server Monitoring (manual update by means of local files).
- **central update**
  The central update service can be carried out after activation in the server configuration via the FirstSpirit Server Monitoring, too (manual update via the update service). The file/s which is/are necessary for the update is/are downloaded directly from the central update service and installed. For this purpose, login data and an internet connection are required. This process can also be automated by the schedule management (automatic update via the update service)

In future, a function will be available for "Enterprise update". With this function several servers can be updated at once (manual update in the corporate network).

*The server update via the central update service is **not** released for the use by end customers in version 5.0. Access for partners is possible under certain conditions for beta tests.*

Special permissions are required for the function described in this chapter. By default, it is available to server administrators.

The release notes on the version to which the system is updated should be read first.

With each software update all modules which are used with FirstSpirit and all web applications must be updated, too. Moreover, services which belong to the installed modules and used web servers, slave servers and the FirstSpirit server itself must be re-started. These steps are mostly

carried out automatically when updating the server (see also *FirstSpirit Documentation for administrators*, Chapter "Server Monitoring" / "FirstSpirit – Control"). Tomcat webserver must be configured correctly for this reason in the server properties (see Chapter 8.8 page 150). In this case, only the self-created modules must then be updated manually.

In the scope of the automatic server update the FirstSpirit server will be stopped automatically and re-started after successful update. The maintenance mode which is also new implemented with FirstSpirit 5.0 can be used for example to inform logged-in users and to prevent new logins of users at the server (see Chapter 8.9 page 152).

8.7.1 Manual updates

The server update takes place via the FirstSpirit Server Monitoring, namely via the "Network" menu item under FirstSpirit / Control or via the menu item "Update" (see Figure 8-7).

Via FirstSpirit / Control / Network:

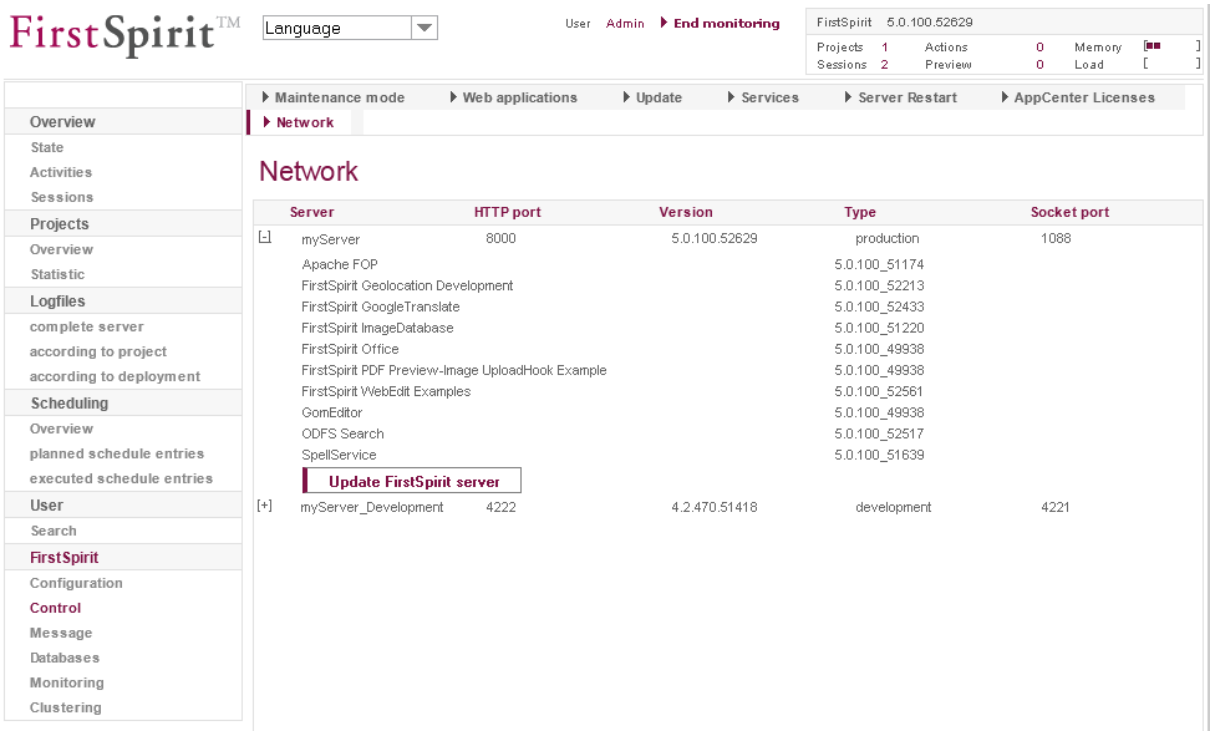


Figure 8-6: Server monitoring – Control – Network

All FirstSpirit Servers of a network are displayed here with the server name, port numbers for HTTP and socket, the FirstSpirit version user and license type as well as all installed FirstSpirit modules (provided available), modules created in-house by the customer are **not** displayed.

The "Update FirstSpirit Server" button can be used to update the respective server.

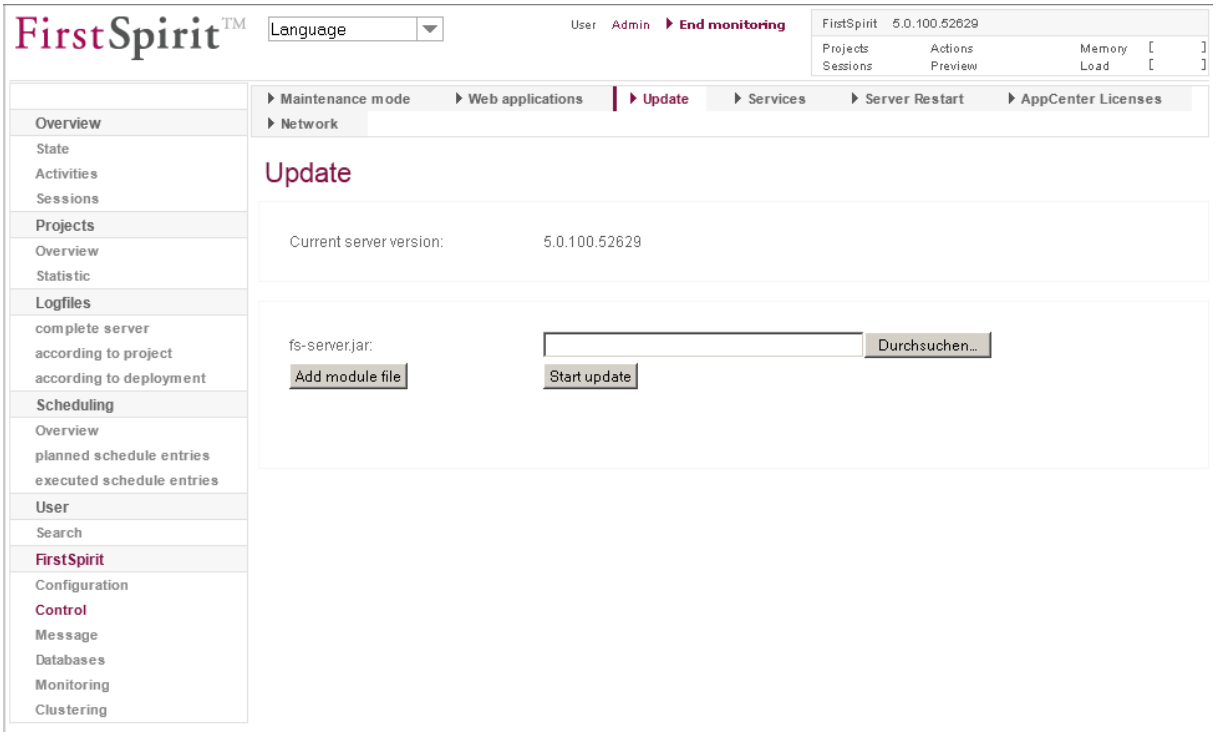


Figure 8-7: Server monitoring – Control – Updating

A page opens, via which the required `fs-server.jar` file and the required module files can be uploaded.

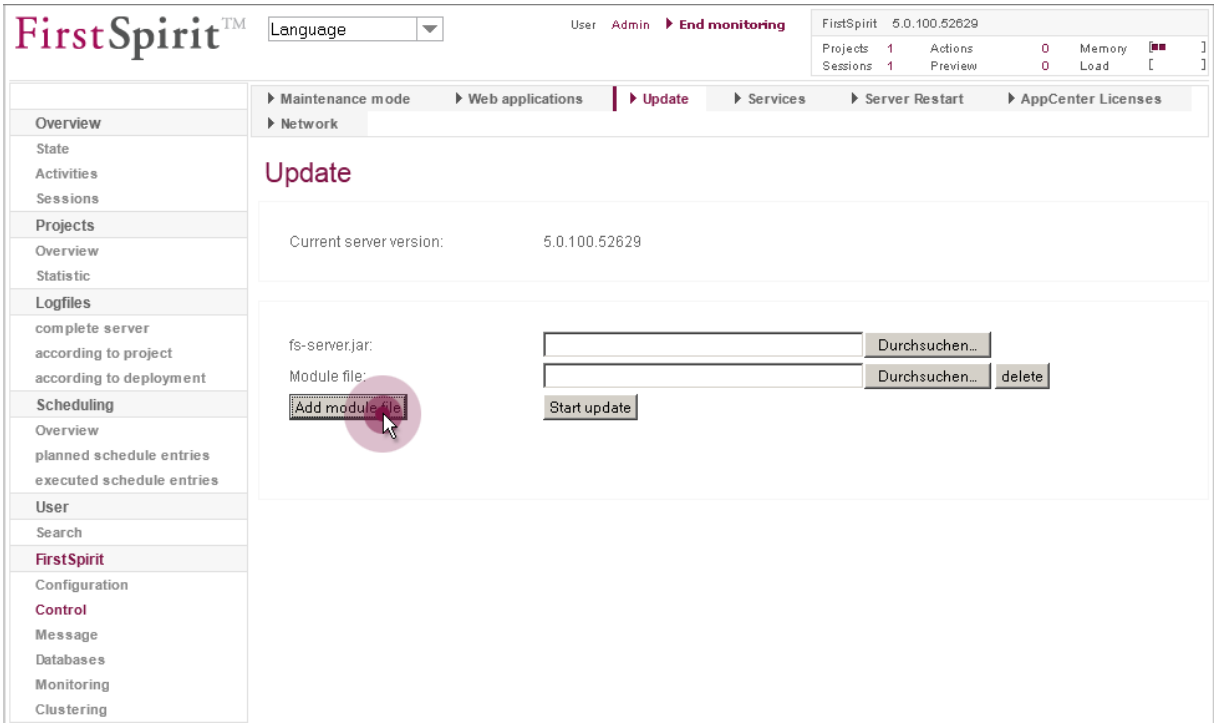


Figure 8-8: Uploading module files

After selecting the files, the server can be updated by pressing the "Start update" button. To this end, after clicking the button, first the files are uploaded, the server is shut down and then restarted automatically.

All Clients logged in to the server should be exited before a server update, to avoid data losses. See also Chapter "Maintenance mode", Chapter 8.9 page 152.

Before shutting down, the files are saved under
`~\server\update\server\lib\` or
`~\server\update\data\modules\update\`
and are loaded from there on rebooting.

This manual server update is active by default. For deactivating the function you can configure this in the FirstSpirit server configuration (file `fs-server.conf`) by means of the following parameter:

`update.push.enabled`: Use this parameter for activating the manual server update (`true`) or deactivating it (`false`). If the parameter is not set the functionality is active.

8.8 Expanded support for Apache Tomcat web servers

If the web server used with FirstSpirit is an Apache Tomcat, FirstSpirit web applications can now be conveniently installed and updated using the FirstSpirit interface. The automatic server update also newly implemented in FirstSpirit 5.0 (see Chapter 8.7 page 146 for more information) furthermore enables automatic updates of installed web applications. In the past, FirstSpirit web applications could only be transferred to the Tomcat web server by way of a WAR (*web application archive*) file and/or scripts (see also, amongst others, *FirstSpirit Documentation for Administrators*, Chapter "FirstSpirit server and project configuration" / "Server properties" / "Web applications" and Chapter "FirstSpirit server and project configuration" / "Project properties" / "Web components").

In FirstSpirit 5.0, the update configuration can be controlled in the server properties (server and project configuration application / "properties"). To do this, the type "Tomcat" needs to be selected in the "Web server" using the "Add" button and the web server control needs to be assigned a name:

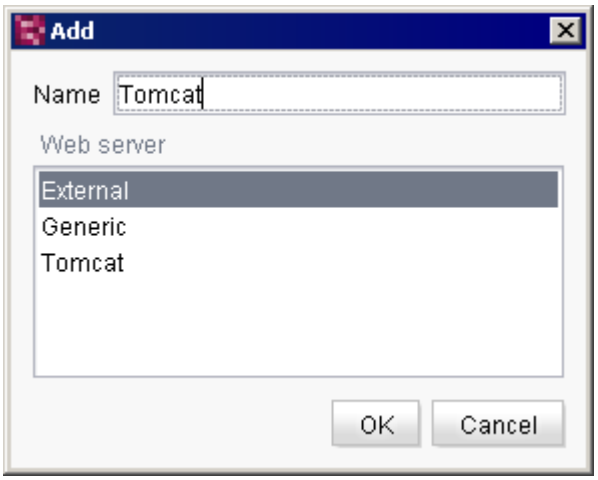


Figure 8-9: Creating a Tomcat web server

The following parameters need to be configured using the "Configure" button in the overview of web server controls:

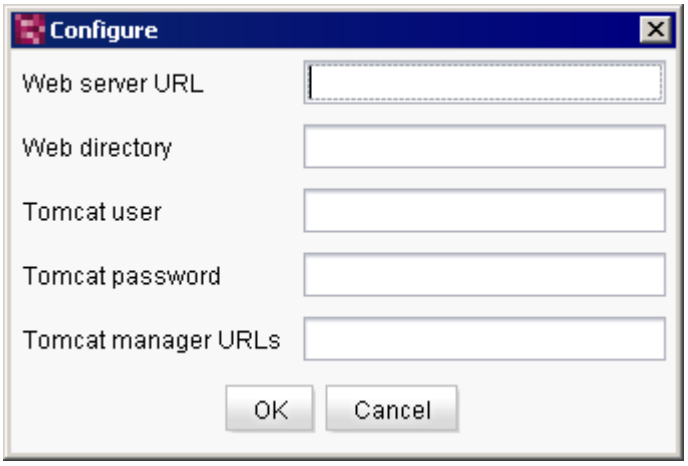


Figure 8-10: Configuring Tomcat control

Web server URL: This is where the URL which is preset when calling needs to be entered, e.g. `http://tomcat:123/fs5webedit`. This URL is for example required for adapting the references for the FirstSpirit web applications on the start page.

Web directory: This is where the path to the web directory of the Tomcat server for using the FirstSpirit web applications (e.g. `fs5staging`) needs to be entered. If the Tomcat web server is for example used for the web application `fs5staging`, the project files will be generated in the specified directory of the Tomcat web server. If it does not exist yet, the web directory entered here will be created in the generation process.

Tomcat user: This is where the user name with access rights to the URL of the Tomcat manager (see below) needs to be entered. User name and password are specified when the Tomcat server is installed and can be looked up in the `tomcat-users.xml` file in the Tomcat web server's `conf` directory.

Tomcat password: This is where the password for the previously entered user name with access rights for the Tomcat manager's URL (see below) needs to be entered. User name and password are specified when the Tomcat server is installed and can be looked up in the `tomcat-users.xml` file in the Tomcat web server's `conf` directory.

Tomcat manager URLs: The Tomcat manager enables the applications to be viewed and administered. It provides both an HTML interface and a text interface. This field is where the URL for the Tomcat manager's text interface needs to be entered, e.g.

`http://localhost:8080/manager/list` for Tomcat 6 or
`http://localhost:8080/manager/text/list` for Tomcat 7

Several URLs can be entered, separated by commas.

This Tomcat web server is available in the "Web applications" and/or "Web components" (analogous to the internal integrated Jetty web server "InternalJetty"):

- Server properties/Web applications
- Project properties/Web components

and must be selected there for the desired web applications, as well as installed and activated by way of the relevant buttons.

For more information on the functions available in these application areas for server and project configuration, see also the *FirstSpirit Documentation for Administrators*, Chapter "FirstSpirit server and project configuration" / "Server properties" / "Web applications" and Chapter "FirstSpirit server and project configuration" / "Project properties" / "Web components".

8.9 Maintenance mode

The newly introduced maintenance mode serves for the following purposes:

- For updating of FirstSpirit servers. To this purpose, the server must be shut down. (However, this is not done automatically by the maintenance mode functionality but has to be done manually in the maintenance mode or in connection with the server update schedule, see Chapter 8.9.1 page 152)
- In order to restrict, eventually also for certain user groups, the access to projects (e.g. during modifications, large updates).

If the maintenance mode is activated, depending on the configuration, no users can log in to the FirstSpirit server or to the selected projects and the open clients are automatically terminated depending on the configuration. The logged-in FirstSpirit users are notified that the server will be shut down and are given the opportunity to save the current changes and log out.

8.9.1 Maintenance mode schedule

In order to activate the maintenance mode, a corresponding server schedule is necessary. It is generated in the application for server and project configuration through "Server" / "Properties" / "Schedule Management" / "Add":

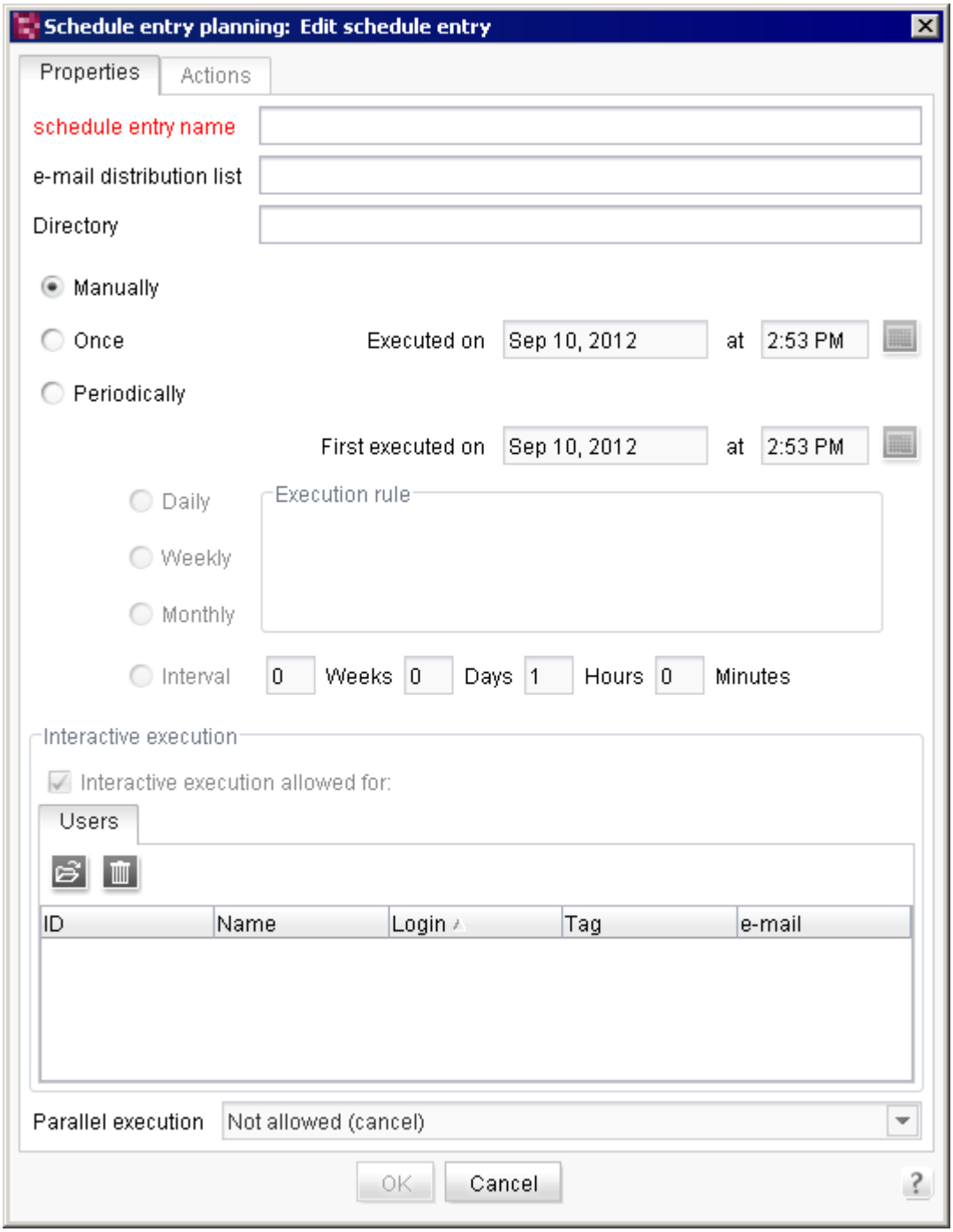


Figure 8-11: New server schedule

If the server or the selected projects are regularly shifted automatically to maintenance mode, the option "Regularly" must be selected and the start time and interval must be determined. If necessary, schedules can be started directly through the schedule overview with the "Manually" option ("Execute" button, see Figure 8-13).

The "Schedule entry name" is used later in the schedule overview (see Figure 8-13) and in Server monitoring (see Figure 8-14).

For further information regarding schedules in FirstSpirit, see FirstSpirit Documentation for Administrators, Chapter *"Server properties" / "Schedule overview" and the following ones)*

The “Maintenance mode” activity is selected on the "Actions" / "Add" register. The following window opens:

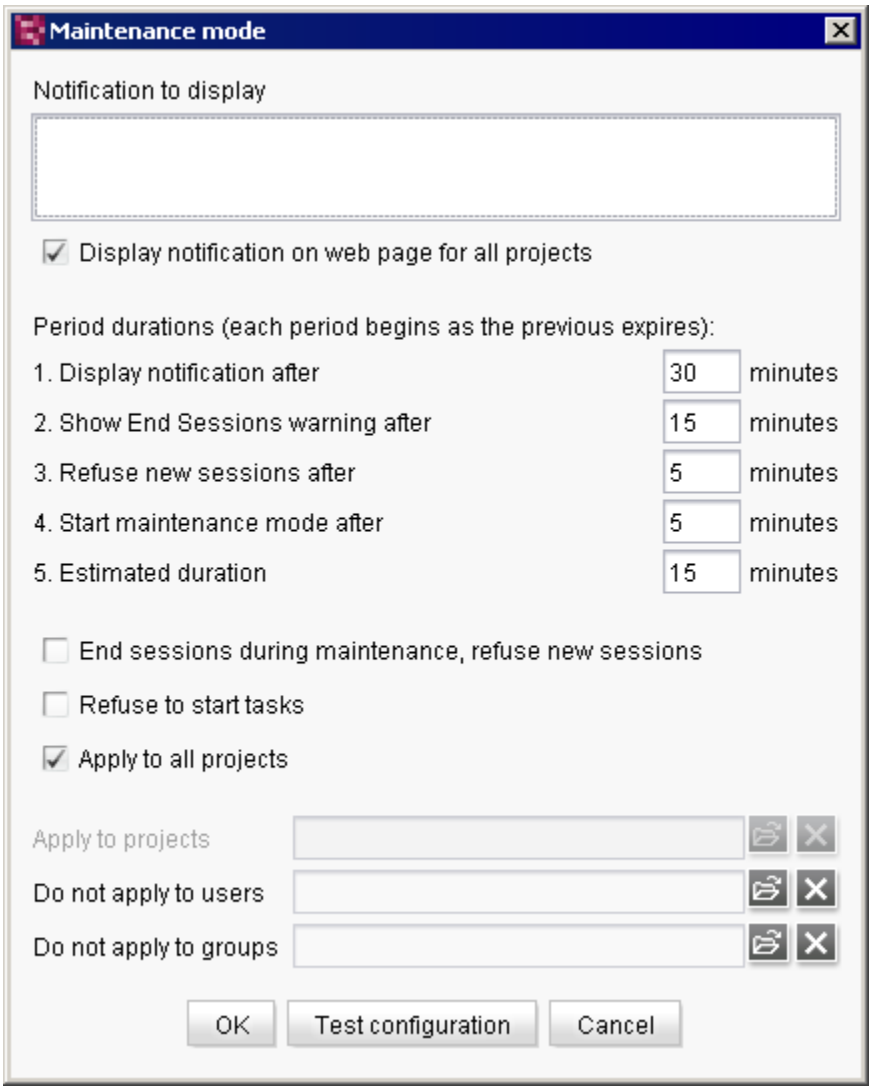


Figure 8-12: Create schedule – maintenance mode (standard configuration)

All data are optional except for steps 4. and 5. It has to be made sure that the selected options fit to each other.

Notification to display: In this case, text can be entered, which is displayed in addition to the

system messages.

Display notification on web page for all projects: If this option is deactivated, the system messages are not displayed on the FirstSpirit start page, but only in the clients.

The maintenance mode consists of several steps, which are always processed automatically in the order that is provided below, as soon as the schedule has started. Here below it can be configured which steps must be executed and at what point in time. Figure 8-12 shows which values are preset by default. Thereby the data in minutes are always related to the preceding action. If the value `0` is entered or if the field is left blank, the respective step is skipped. The generated system messages depend then on whether the schedule is valid for the entire server (option "Apply to all projects", see below) or only for one or several projects (option "Apply to projects", see below).

1. Display notification after: After the number of minutes provided here, the system message “FirstSpirit server maintenance will be performed in x minutes. Planned duration: about y minutes” is generated as an advance warning and eventually the text entered in the "Notification to display" field is displayed. Thereby `x` is the number entered in this field and `y` is the number entered at the "Estimated duration" option.

2. Show End sessions warning after: After the number of minutes provided here, the system message "The FirstSpirit server will become unavailable due to maintenance at Oct 2, 2012 11:58:46 AM. Planned duration: about y minutes. Please end your session" is generated and eventually the text entered in the "Notification message" field is displayed. Now the logged-in users have the possibility to save their work and then log out. Thereby the date and time is the starting point of the maintenance mode schedule plus the duration of steps 1. to 4.

3. Refuse new sessions after: If the option "End sessions during maintenance, refuse new sessions" (see below) is activated, after the number of minutes provided here it is not possible anymore to log in to the server or to the selected projects.

4. Start maintenance mode after: The maintenance period begins after the number of minutes provided here. If the option "End sessions during maintenance, refuse new sessions" (see below) is activated, the clients, which are still logged in, are logged out with the message "The FirstSpirit server | This project is unavailable due to maintenance. Your session has been ended". Depending on the setup, the following system message is displayed on the start page and in the client: "The FirstSpirit server is in maintenance mode since Oct 2, 2012 11:58:46 AM. Planned duration: about y minutes." Now the maintenance works can start and, if necessary, the FirstSpirit server can be shut down

5. Estimated duration: Here it can be specified how long (in minutes) the maintenance works

will continue. After this period of time expires, the following system message is generated automatically: "FirstSpirit server | Project maintenance has ended" and then it is again possible to log in to the server. If during a running schedule it is established that a longer period of maintenance mode is necessary than the one that has been preset, the schedule must be ended (prior to the end of this step) manually (see Chapter 8.9.2 page 157).

It can be seen in the Server monitoring in which step a started maintenance mode schedule is located (see Chapter 8.9.2 page 157).

End sessions during maintenance, refuse new sessions: If this option is activated, all sessions of the server or of the selected projects during the maintenance period are ended. If the option is deactivated, no sessions are ended. In order to allow new sessions, in addition, “Refuse new sessions after" must have the value of `0` in step 3.

Refuse to start tasks: If this option is activated, schedules, which are in the maintenance time period, are cancelled.

Apply to all projects: If this option is activated, the schedule is valid for all projects on the server. If the option is deactivated, the desired project(s) can be selected under "**Apply to projects**".

Apply to projects: The project(s), for which the order must be valid, can be selected here.

Do not apply to users: A given user, who can log in to the FirstSpirit server or to the selected projects in maintenance mode, can be selected here. The server administrator can log in anyway.

Do not apply to groups: Groups, which can log in to the FirstSpirit server or to the selected projects in maintenance mode, can be selected here.

By clicking on "OK", the order is saved and then appears in the schedule overview:

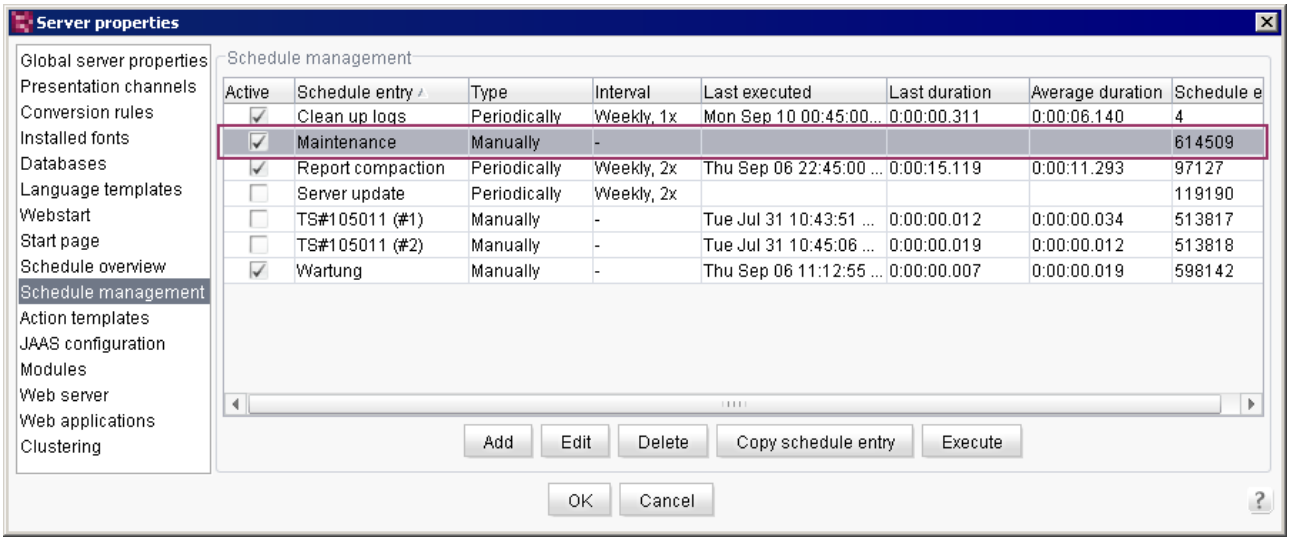


Figure 8-13: Schedule overview

The schedule will start at the preset time, or it can be started directly with the "Execute" button. Maintenance mode schedules which have been already started can be stopped via the FirstSpirit Server monitoring (see Chapter 8.9.2 page 157). Maintenance mode schedules which have not been started yet can be stopped in the order itself by changing to "Manually".

8.9.2 Managing the maintenance mode via the Server monitoring

With the FirstSpirit server monitoring ("FirstSpirit" / "Management" / "Maintenance mode" area), it is possible to

- start maintenance mode schedules
- view the current status of maintenance mode schedules, and
- stop current maintenance mode schedules

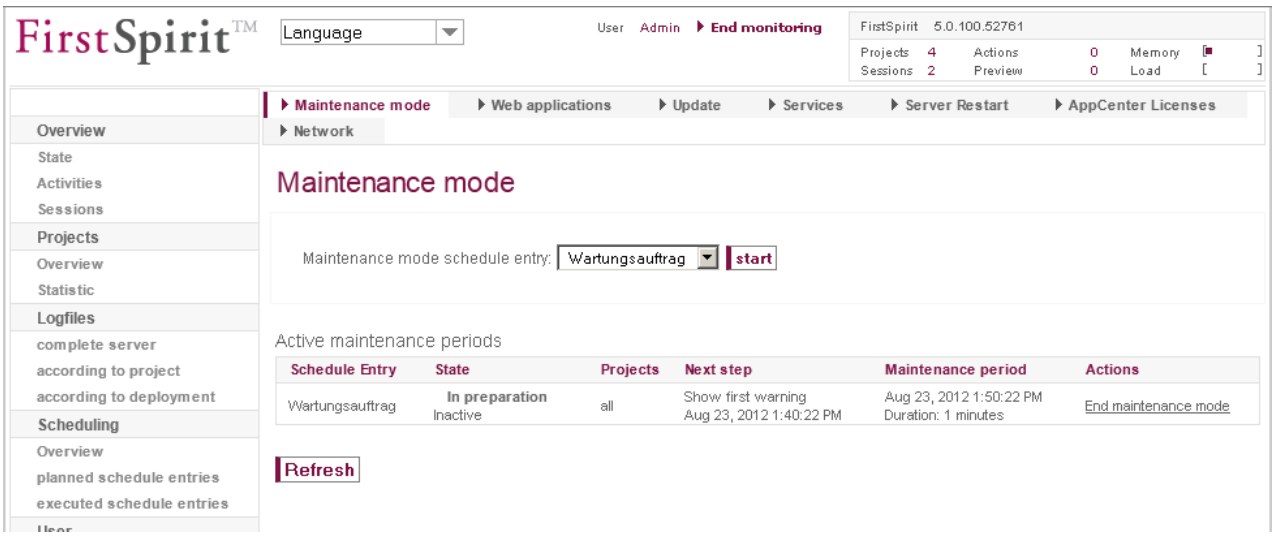


Figure 8-14: Maintenance mode in Server monitoring

start: If a maintenance mode schedule has been created (see Chapter 8.9.1 page 152), it can be selected from the "Maintenance mode schedule entry" drop-down list and started with the “start” button. A "Maintenance mode started" message is generated. A given schedule can be started also repeatedly at the same time by using this button.

Planned maintenance periods: Planned maintenance mode schedules are displayed here with the settings which were specified in the server properties.

Active maintenance periods: Running maintenance mode schedules are displayed here with the settings which were specified in the server properties.

State: This column shows the current status of the schedule (press the “Refresh” button for a current display). Thereby, depending on the settings in the schedule, the statuses are shown in the following order:

- *In preparation* | *Inactive:* The schedule has started, but there are not yet any effects for the user.
- *In preparation* | *Show first warning:* In this status, the message “FirstSpirit server maintenance will be performed in x minutes. Planned duration: about y minutes” is displayed in the clients and, if necessary, on the start page as advance warning. See the option "Display notification after" in the schedule (Chapter 8.9.1 page 152).
- *In preparation* | *End sessions warning:* In this status, the message "The FirstSpirit server will become unavailable due to maintenance at Oct 2, 2012 11:58:46 AM. Planned duration: about y minutes. Please end your session" is displayed in the clients and, if necessary, on the start page. See the "Show End sessions warning after" option in the schedule (Chapter 8.9.1 page 152).
- *In preparation* | *Refuse sessions:* In this status, no logins are possible anymore. See the

"Refuse new sessions after" option in the schedule (Chapter 8.9.1 page 152).

- *Active | Maintenance:* In this status, the maintenance mode is active. Now the maintenance works can start and, if necessary, the FirstSpirit server is shut down. See the "Start maintenance mode after" option in the schedule (Chapter 8.9.1 page 152).
- *End:* In this status, it is again possible to log in to the server. See the "Estimated duration" option in the schedule (Chapter 8.9.1 page 152).

Projects: It is displayed in this column which projects are affected by the maintenance mode schedule ("Apply to projects" option in the schedule, see Chapter 8.9.1 page 152).

Next step: It is displayed in this column which step is the next one and when it starts. The steps are described under "State" (see above).

Maintenance period: It is displayed in this column when the maintenance period starts (it is obtained from the schedule start time and the time data for the steps 1. to 4.) and what its duration is ("Estimated duration" option in the schedule, see Chapter 8.9.1 page 152). Depending on the configuration of the options "Refuse to start tasks" and "End sessions during maintenance, refuse new sessions", the schedules, which are within the maintenance time period, are cancelled and no logins on the server or in the selected projects are possible during the maintenance period.

End maintenance mode: With a click on this link, the corresponding schedule is ended. In Server monitoring, the "Maintenance mode stopped" message is displayed. In the clients and on the start page, the message "FirstSpirit server maintenance was completed." is displayed.

Refresh: With this button, the view can be updated.

8.10 Automatic reporting of software errors

*This feature is **not** released for the use by end customers in version 5.0. Access for partners is possible under certain conditions for beta tests.*

The implementations begun with FirstSpirit Version 4.2 with regard to improved and easier error reporting have been picked up again for Version 5.0. The aim is to achieve software-assisted recording and management of errors. In this way, a decisive contribution can be made to faster analysis and correction of errors and therefore to increased software quality.

The infrastructure for Internet-based error reporting has now been created with Version 5.0. Error

reports are transmitted to e-Spirit via a web interface and apart from the Client's error reports also contain server errors (e.g. errors during generation) and monitoring indicators (e.g. GC Time and Heap).

8.11 New browser engine

Up to now, the Mozilla Firefox in the Version 3.6 has been used for the integrated preview. With FirstSpirit 5.0 the Mozilla Firefox in Version 15 Beta is available in addition.

If this engine can be selected in the project or if it is the only available engine in the project can be set in the Project properties (Application for the Server and project configuration / Client applications / Browser engine):

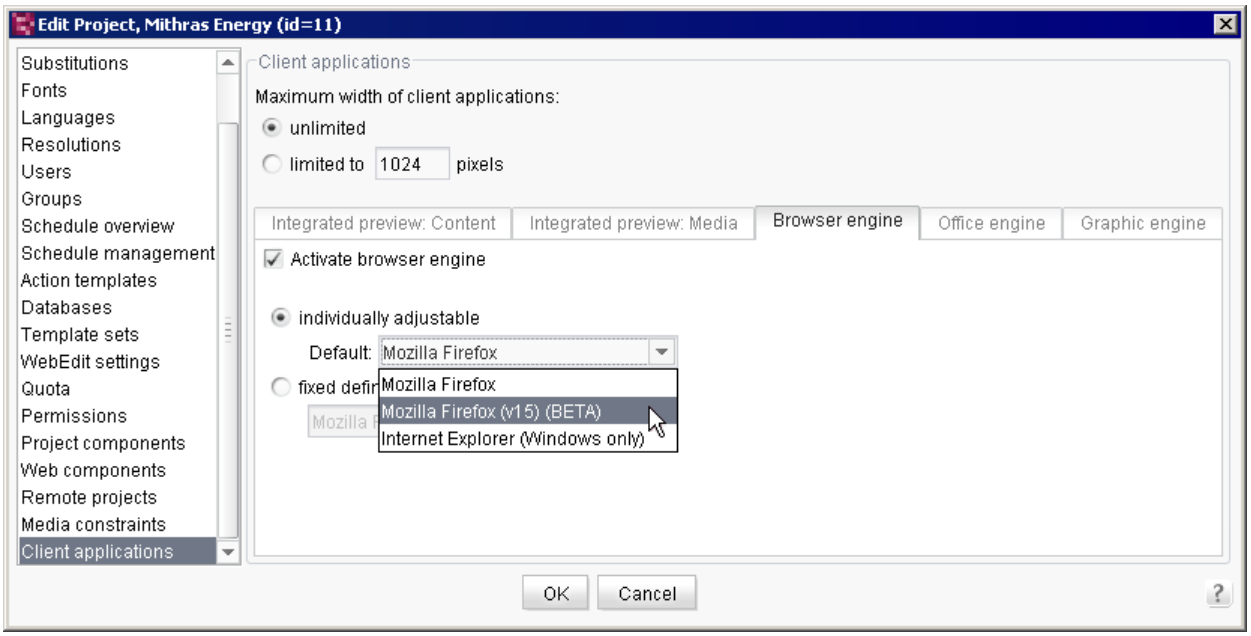


Figure 8-15: Selection of the Mozilla Firefox 15 in the project properties

For further information about configurational options on this tab see also FirstSpirit Documentation for administrators, Chapter "Client applications".

This engine is for the time being in the state BETA and is, for this reason, not officially released.

The Mozilla engine in the Version 3.6 will be omitted in FirstSpirit Version 5.1.

9 New / changed functions in modules

9.1 FirstSpirit Multisite Management

In the "Multi-site management" area, the license-dependent functionalities "FirstSpirit CorporateContent" and "FirstSpirit ContentTransport" are available.

The function **FirstSpirit CorporateContent** represents a further development of the previous function "Package management", with which the templates and the contents can be distributed automatically among different projects on a given server. In this way, contents can be comfortably reused across projects with FirstSpirit. An important aspect of the assembling of the packages is that all depending objects must also be comprised in the package. In FirstSpirit Version 5.0, in the CorporateContent area, the usability for the user is significantly improved (in the assembling of the packages).

In order not to generate migration costs in the existing CorporateContent (package management) projects, the basic structures for the subscription and package relationships, as well as the automated unroll mechanisms, were maintained compatible. The most important functional expansion is the possibility for creation of packages based on the latest released version.

The recently added **FirstSpirit ContentTransport** function is a completely new development, and it now can transport, along with templates, also contents (the so-called "Features") between projects through the server boundaries. This functionality offers a comfortable user interface for the compilation and updating of the features. Unlike the CorporateContent packages, the features must not be connected, i.e. not all referenced object have to be contained in one feature.

In version 5.0, the ContentTransport replaces completely the "Model updating" (also "Template updating") functionality, which belongs to the product core and is known from earlier FirstSpirit versions. With Template updating it is possible to distribute templates (but not contents!) manually between different projects (potentially also between different servers).

Now with ContentTransport it is possible to display much better scenarios, in which new functions, consisting of templates AND contents (therefore, structures, data, media and datasets), can be transferred from a development system (D) to a quality assurance system (Q), so that the function can be tested there. After the successful test, the function, originating from D, is transferred further to the productive system (P). In order to automate the transfer of the

features between the DQP systems, an API is available.

When this functionality is used, it must be paid attention to the following constraints:

- **Rights** can not be transported; they must be set up manually in the target project. However, this must be done only once; if necessary, the assignments are kept in the next updating processes.
- **Project properties**, which have been set up in the application for the server and project configuration, are not transportable.
- **Work process statuses** are not transported.
- **Metadata** are transported, but they may be incompatible with the selected metadata template depending on the configuration in the target project.
- **Delete operations** are transported when the root node is contained in the package **and** there are no references in the target project. If the target project contains references, the object is moved to "Lost&Found".
- If the target project contains objects (e.g. templates) in **processing mode**, the import procedure can not be performed for these objects. Therefore, it has to be made sure that there are no objects in processing mode!

In the future, the "ContentTransport" area is expanded with functions, which (analogically to the existing CorporateContent) permit such a configuration of the automatic distribution of features to different projects also through the server boundaries, so that then CorporateContent scenarios could be converted to the ContentTransport method also without the use of an API.

The modules "FirstSpirit CorporateContent" and "FirstSpirit CorporateTransport" are subject to licensing and need the same license. For detailed information about this module, see also the corresponding module documentation.

If there is a valid license, the functionality "CorporateContent" is accessible in FirstSpirit JavaClient from the vertical toolbar in the left side area of JavaClients through the icon  , and the functionality "ContentTransport" – through the icon  .

Here it is possible to create and process packets or features and view information about the contained objects:

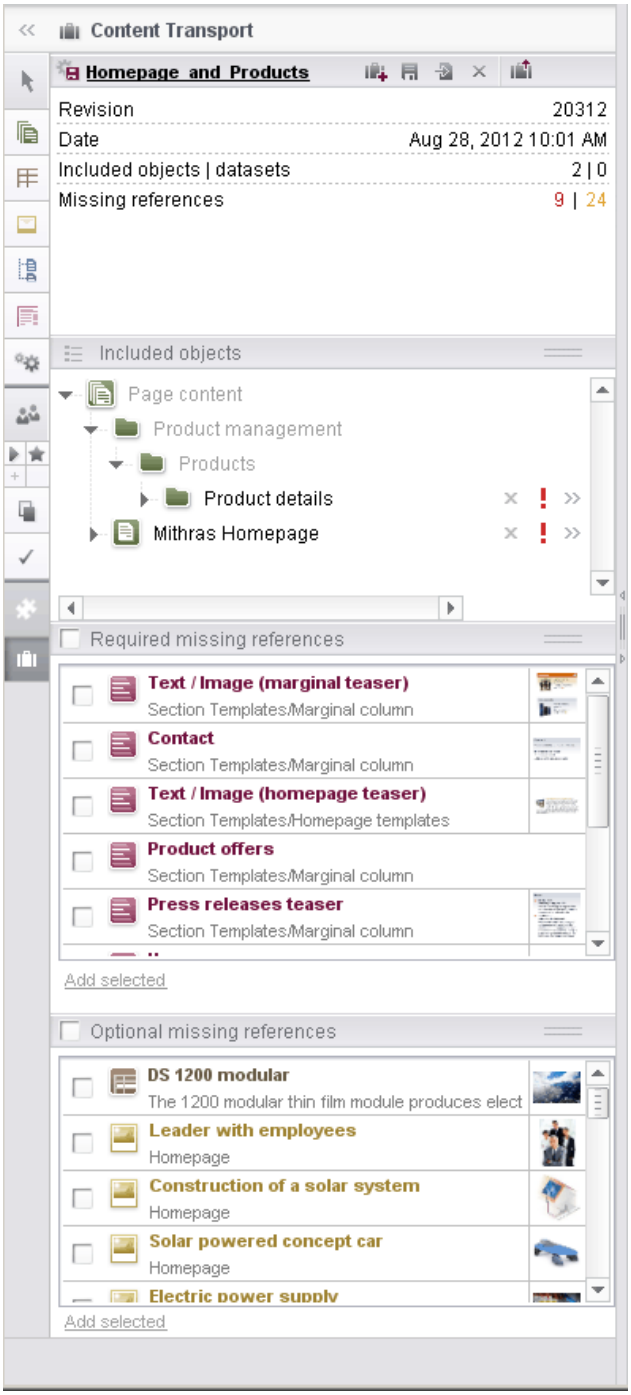


Figure 9-1: Information about a feature

A graphical representation in the AppCenter area permits a flexible view of the hierarchical structure and the dependencies of the dependent objects:

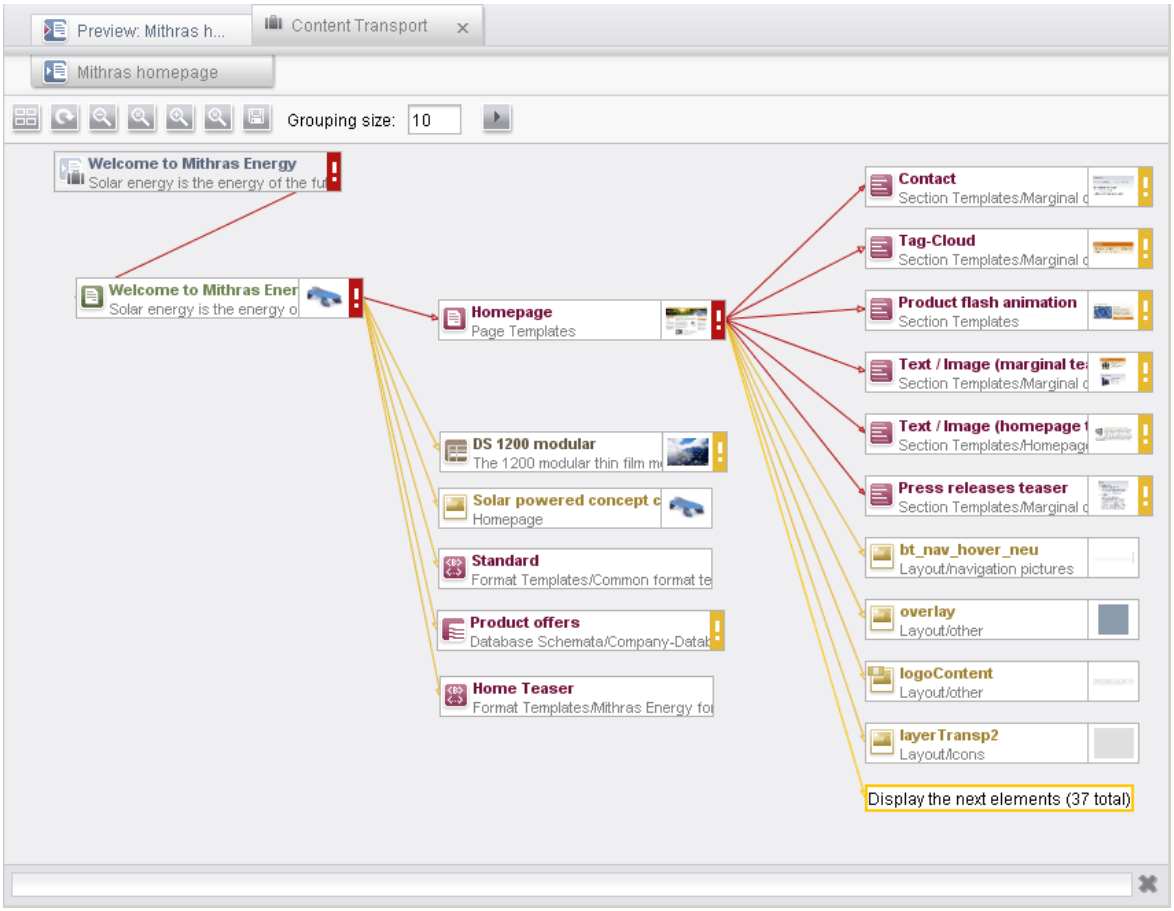


Figure 9-2: Graphical representation in the AppCenter area

For further information about the "ContentTransport" functionality, see Chapter *2.3 page 14.*

9.2 FirstSpirit BasicSearch (formerly "Search")

This module is subject to licensing. Please also see the module documentation for more detailed information on the module.

Owing to new security mechanisms indexing of the generated project content by a search engine is no longer possible without authentication information. In this case the configuration of the "SEARCH" module must be adjusted and the necessary login information added (for further information see Module Documentation for *FirstSpirit SEARCH*).

10 Appendix

10.1 Changes in the software behavior

- **Software interface JavaClient:** In the tree structure of Version 5.0, the Stores are no longer provided in bars and/or icons at the bottom of the tree structure, but instead available in a vertical tool bar at the far left of the JavaClient (see also Chapter 6.1.1 page 37). The labels of the root nodes have been changed:
  German:
  -  Inhalte
  -  Datenquellen
  -  Medien
  -  Struktur
  -  Vorlagen
  -  Globale Einstellungen

  English:
  -  Page content
  -  Data sources
  -  Media
  -  Site structure
  -  Templates
  -  Global settings
- **Modified operating concept:** The use of dynamic forms (see Chapter 7.4 page 103) as from FirstSpirit Version 5.0 for example permits changes in the handling of invalid contents that would potentially be incompatible with earlier versions of FirstSpirit: While earlier versions of

FirstSpirit allowed pages with invalid internal references, for example, to be released, this can now be prevented by defining suitable rules.

- **Integrated preview – maintaining the position when saving:** The element provided with the content highlighting frame in the integrated preview of the JavaClient because it is currently selected in the workspace and/or tree structure will now always remain in the visible area of the preview and/or be moved to this visible area. This helps to ensure that the element currently being edited by the editor is always shown in the integrated preview, also after saving, without the editor needing to scroll to the output location in the preview first. This is particularly useful for elements located relatively near the end of "long" pages.
- **Export/import of individual nodes:** The functions "Export" and "Import" are now available in the context menu for most node types in the FirstSpirit JavaClient. This function is only available to administrators. It can be used to export the selected node to the hard disk with all the information required to re-import it to another project again later. The format used has been drastically changed in Version 5.0. The import of earlier V4.2 context menu exports is still possible. The developers have aimed to provide backward compatibility that would principally allow context menu export from V5.0 to V4.2, but this is not a guaranteed product feature!
- **Restore deleted objects:** If a node of the tree structure in JavaClient has been deleted, the node can be restored mostly through the function "Restore deleted objects" in the context menu of a higher-level node. The dialog for selecting the position, where the deleted object has to be inserted, has been changed:

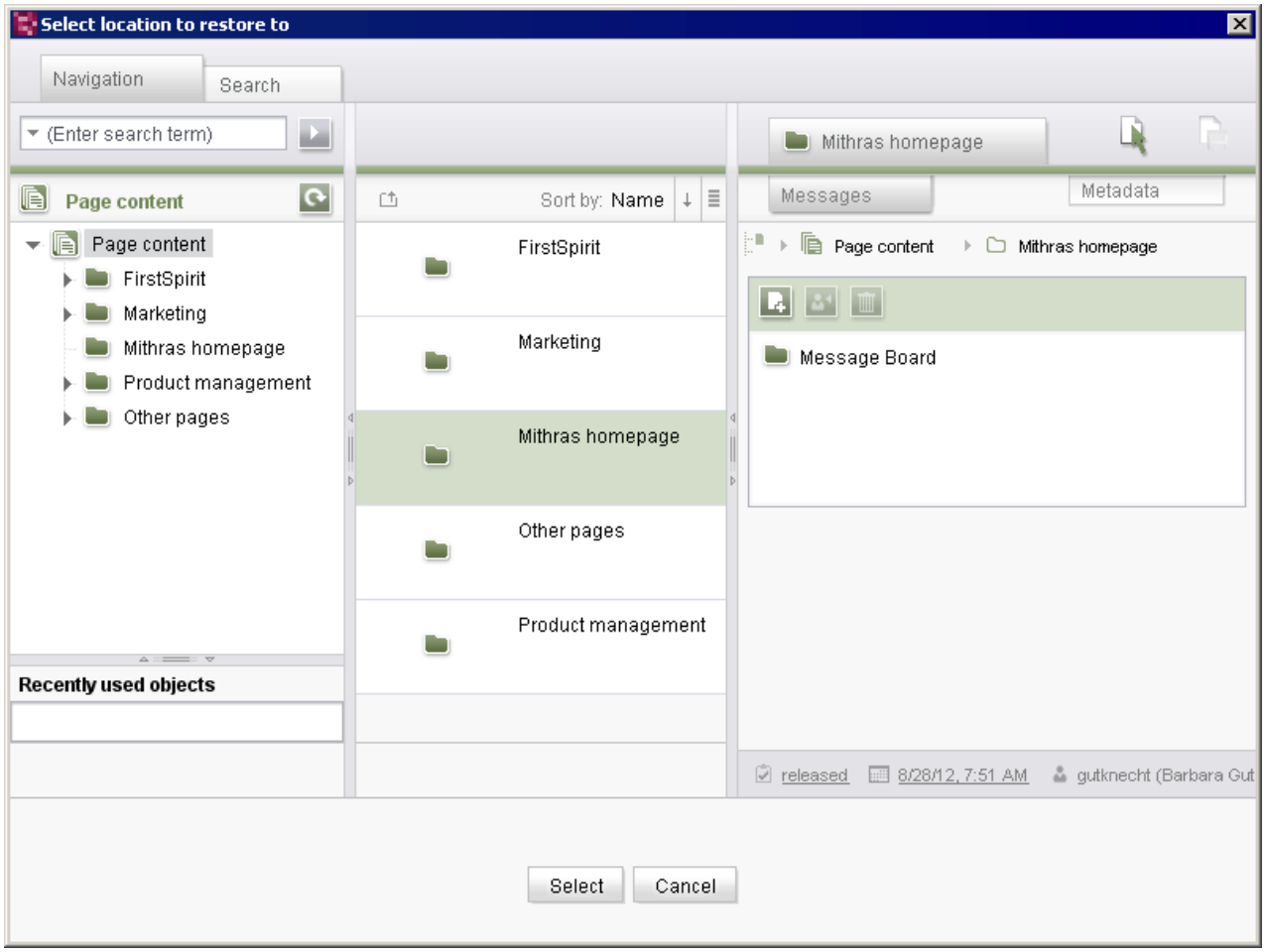


Figure 10-1: Select the restore position

As a rule, a subfolder is selected in the dialog. If the root node of a given admin must be selected, the subfolder marked in the middle column must be deselected with Ctrl and click on the subfolder.

- **Stored URLs in the object properties:** The "Information" dialog available by pressing ALT + P or selecting the context menu option "Show properties" under "Extras" has been supplemented by the new tab "Stored URLs":

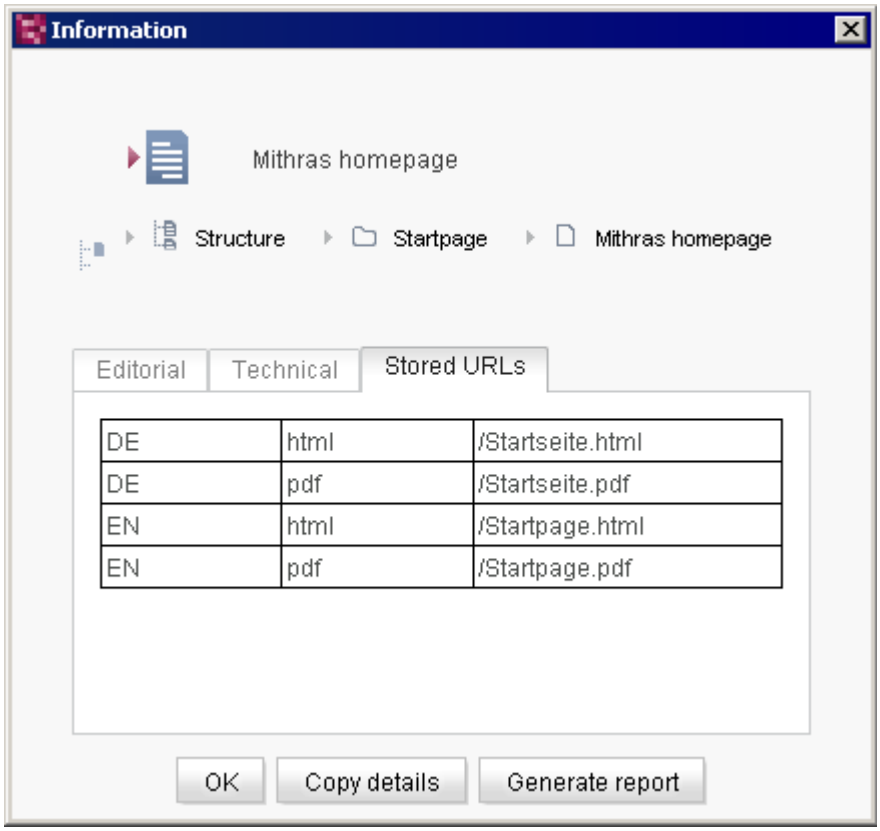


Figure 10-2: Properties of a page reference – Stored URLs

This is where the URLs stored for page references, menu levels and media can be viewed and reset by selecting the context menu option "Reset stored URLs" under "Extras". See also Chapter 6.1.15 page 75 and Chapter 8.5 page 125.

- **Saving of data records:** In contrast to earlier versions, empty data records can now be saved because of a change in the software.
- **Search in data sources:** A search in data sources, which can be executed through the link above the data source view, does not open anymore the dialog with the different search types (full-text search, available search queries, simple search and expanded search), but the text to be searched for can be entered directly in the search field, which is opened with a click on the link:

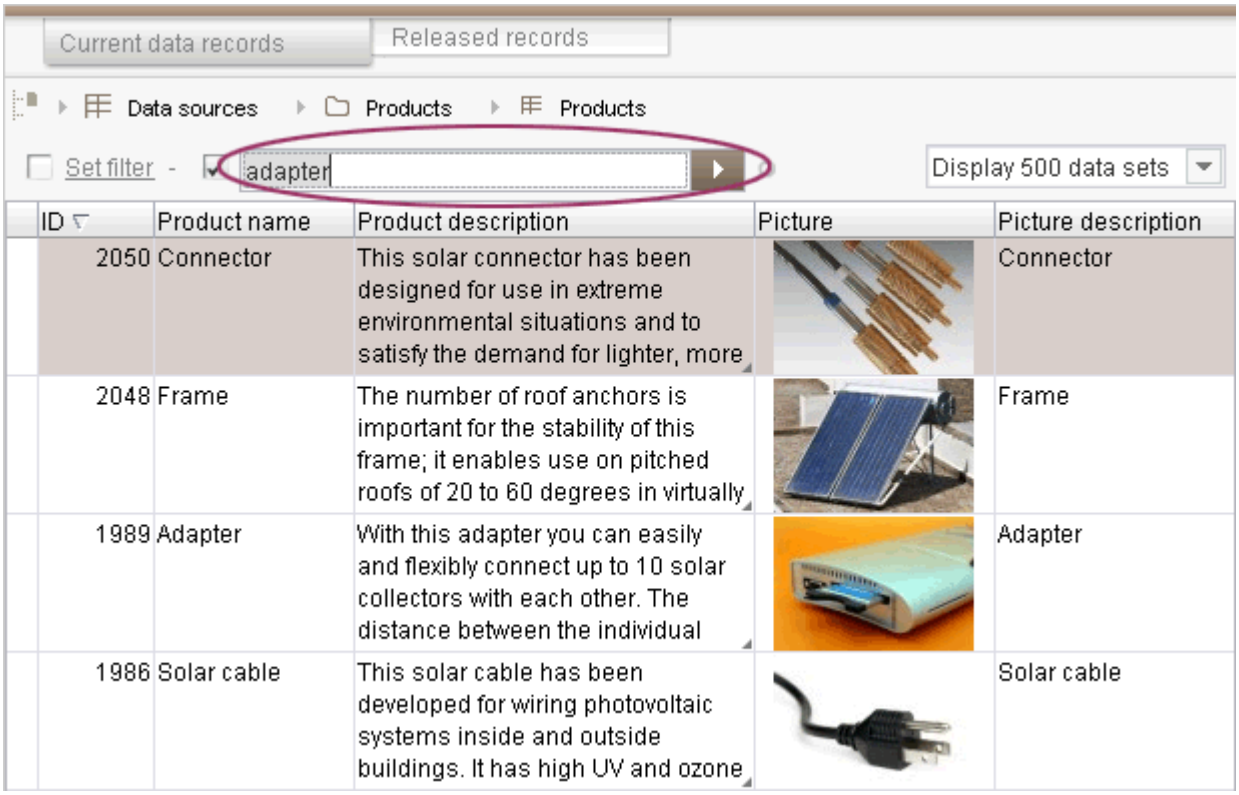


Figure 10-3: Search in data sources

The search string is highlighted in color in the detailed display of the found datasets.

- **Line numbering:** In Version 5.0, line numbers are displayed by default at many places of the template development (JavaClient and server and project configuration application) for easier reading. They can be hidden/displayed in the JavaClient by pressing the CTRL + L keyboard shortcut.
- **Keyboard shortcut for showing dependencies:** As an alternative to the context menu option "Extras" / "Show dependencies", the keyboard shortcut CTRL + R can now also be used to show dependencies from nodes in the tree structure (reference graph).
- **PicMonkey graphics engine:** The "Picnik" graphics engine for editing images in the Media Store of the JavaClient (menu "View" / "Graphics engine" / "Easy image processing (Picnik)" and/or the server and project configuration application at "Project properties" / "Client application" / "Graphics engine") has been replaced by the "PicMonkey" graphics engine because the www.picnik.de service has been discontinued. See also information on "Picnik" in the *FirstSpirit Release Notes for Version 4.2R4*.
- **"Mithras Energy" demo project:** The standard demo project included in FirstSpirit has been revised and adapted to the givens and requirements in FirstSpirit Version 5.0.
- **Reference names when uploading media:** Whereas the reference name could be specified

in earlier versions of FirstSpirit directly when uploading media to a project, for example using input components such as CMS_INPUT_PICTURE or FS_REFERENCE, this is no longer possible in Version 5.0. The reference name will be derived from the file name and can only be changed after the upload process – given the right settings.

- **Execution of scripts:** Scripts which are started in the JavaClient (e. g. by means of the context menu "Extras") can now no longer block the JavaClient because they are no more executed in the EDT (`java.awt.EventDispatchThread`).
- **Renaming of web applications and directories:** Owing to the major release change from FirstSpirit Version 4.x to Version 5.0, all FirstSpirit web applications have been renamed. As a consequence, the directories have changed too: `~fs4\web\fs4root`, `~fs4\web\fs4preview`, `~fs4\web\fs4staging` and `~fs4\web\fs4webedit` for example, turned into `~fs5\web\fs5root`, `~fs5\web\fs5preview`, `~fs5\web\fs5staging` and `~fs5\web\fs5webedit` in FirstSpirit Version 5.0.
- **Storage space for the client's virtual machine:** Whereas the following values (in MB) used to be selectable for the storage space available to the client's virtual machine (e.g. FirstSpirit Start Page / "Connection settings" / "Storage"): 128, 256, 512 or 1024, these have now been raised to 512, 768, 1024 and 1536 in FirstSpirit 5.0.
- **JMX configuration**: The JMX configuration is now implemented by means of an `fs-server.conf` file instead of the `fs-wrapper.conf` file used previously. The parameters for this are described in the *FirstSpirit Handbook for Administrators,* section "fs-server.conf" / "Area: JMX".
- **Path creation in the generation process:** The "Infix URLs" method is marked as legacy. Search machine optimization variants (SEO) have furthermore been added to the methods "Default", "Multiviews" and "Infix". In generation schedules, these variants are supplemented by an "(SEO)" and ensure the storage of the generated URLs. See also Chapter 8.5 page 125. The first generation process in SEO mode can take some time.
- **Conversion rules:** The server properties (server and project configuration application / Server / Properties / Conversion rules) now permit the definition of conversion rules that can then be selected in the project for format templates, for example. As from FirstSpirit 5.0, only one conversion rule is included in the standard scope, namely "Convert HTML" with the following contents:

```
[convert]

0x3c="&lt;"
0x3e="&gt;"
0x22="&#34;"
0x26="&amp;"
0x27="&#39;"
```

FirstSpirit demo projects, e.g. "Mithras Energy", have been adapted to this conversion rule. Individual conversion rules can naturally still be created. Conversion rules which exist already since earlier FirstSpirit versions remain and are not deleted. Please see the *FirstSpirit Documentation for Administrators*, Chapter "Conversion rules", *FirstSpirit Handbook for Developers (Part 1: Basics)*, Chapter "Format templates" and the FirstSpirit online documentation, Chapter "Template development" / "Format templates" / "Conversion" for more information.

- **Mandatory fields in the WebClient:** Mandatory fields in the WebClient 5.0 which have previously been identified by an asterisk must be mapped by means of so-called "dynamic forms" (see Chapter 7.4 on page 103) in FirstSpirit Version 5.0.
- **FirstSpirit modules – Apache FOP:** In FirstSpirit 5.0, the version used for the "Apache FOP" module has been updated to 1.0. If error messages such as

```
Font "Symbol,normal,700" not found. Substituting with "Symbol,normal,400".
```

should be shown, they can be circumvented by defining the parameter `font-family` with the default font of the document in `fo:root`, e.g.:

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format" font-family="Helvetica">
```

10.2 Dropped functions in FirstSpirit Version 5.0

The following functions have been dropped as from FirstSpirit Version 5.0:

- **"FS_" input components:** The input component types replaced by the new "FS_" input components are deprecated with Version 5.0. This means that these "old" input components can still continue to be used in the initial released version of FirstSpirit 5.0, but they will no longer be further developed and only maintained to a limited extent (only limited bug fixes will be made) and will be removed completely in a later version. For WebClient, these input components are already no longer supported in Version 5.0. (See also Chapter 7.3.1 page 101.)
- **Replacement of template updates with "Content Transport":** The function "Template update" familiar from version 4.x has been reimplemented in its entirety in Version 5.0 and turned into a new function named "Content Transport" that offers a significantly expanded

range of options with a focus on the realization of scenarios requiring the application of development, test and productive systems (DQP). In this context the Content Transport function has also been integrated in the FirstSpirit module "CorporateContent". The significantly expanded functions of the Content Transport replace the template update of version 4.x, which is being dropped as from version 5.0 (see also Chapter 9.1 page 161 and Chapter 6.1.16 page 86).

- **Link templates:** Non-generic links (link types "internalLink", "externalLink" and "contentLink") are no longer being officially supported as from FirstSpirit Version 5.0 and should be converted to the new, generic link format while still in version 4.x using the context menu function "Convert link template" in the link configuration. This context menu function is no longer available in version 5.0. If non-generic link templates are still being used in a project, the generation process on FirstSpirit servers of version 4.2R4 will be attended by "INFO" messages that are logged in the log file and contain the reference name of the respective link template:

```
INFO 10.07.2012 14:01:24.618 {seID=2567}
(de.espirit.firstspirit.store.access.templatestore.LinkTemplateImpl): usage
of deprecated old link template 'textlinkinternal.standard' (project=422823,
id=493962)
```

In Version 5.0 FirstSpirit servers, "WARN" messages will now be logged in the log file:

```
WARN 10.07.2012 14:05:31.318 {seID=492003}
(de.espirit.firstspirit.store.access.templatestore.LinkTemplateImpl): usage
of deprecated old link template 'textlinkinternal.standard' (project=422823,
id=493962)
```

These messages can be used to identify instances of the deprecated link template being used.

- **Input component CMS_INPUT_TEXTAREA:** The `htmlMode` parameter is no longer being supported as from FirstSpirit Version 5.0. Usages of this parameter from earlier FirstSpirit versions are deleted when editing respective templates.
- **Module – Apache FOP:** The support for **Apache FOP** (formatting objects processor) in FirstSpirit, which for example allows FirstSpirit contents to be output in PDF format, is realized by way of a module. The previously supported version FOP 0.20.5 is now deprecated as from FirstSpirit Version 5.0. This means that the "Apache FOP v0_20_5" module will no longer be included in the installation and that there will be no more bug fixes for it. The "Apache FOP" module can be used instead (see also Chapter 10.1, entry "FirstSpirit modules – Apache FOP").

10.3 Announcements for future versions

- **System requirements and third-party software:** Support of the following data base, operating system, JDK and HTTP-/ application server versions is planned for FirstSpirit versions following the initial release of FirstSpirit 5.0:
    - Operating system (client-side): Microsoft Windows 8, Mac OS X 10.8
    - Operating system (server-side): Microsoft Windows Server 2012
    - Java environment (client-side): OpenJDK 7
    - Java environment (server-side): OpenJDK 7, Azul VM
    - HTTP-/application server: Apache HTTP Server Version 2.4
    - Data bases: Microsoft SQL Server 2012, IBM DB2 10

- The **Mozilla browser engine** in Version 3.6 which is used for displaying content in the Integrated preview at the moment, will be omitted with FirstSpirit Version 5.1. See also Chapter 8.11 page 160.
- With respect to the JavaClient, the support for projects that do not use **releases** (server and project configuration application, "Options", deactivated option "Use release") will be discontinued in version line 5.x.
- The following **input components** are being marked as deprecated in FirstSpirit Version 5.0 for the JavaClient and will be dropped in a future version:
    - CMS_INPUT_CONTENTAREALIST
    - CMS_INPUT_CONTENTLIST
    - CMS_INPUT_FILE
    - CMS_INPUT_LINKLIST
    - CMS_INPUT_OBJECTCHOOSER
    - CMS_INPUT_PAGEREF
    - CMS_INPUT_PICTURE
    - CMS_INPUT_SECTIONLIST
    - CMS_INPUT_TABLIST

  They are being replaced by the input components with the prefix *FS_*, which have been newly introduced in version line 4.2 and released with FirstSpirit 5.0:
    - FS_LIST
    - FS_DATASET
    - FS_REFERENCE

  See also Chapter 7.3 page 101 for more information.

- FirstSpirit Version 5.1 will provide fundamental restructuring in the internal implementation. These changes will not affect the released API, but the use of FirstSpirit implementations that are not part of the released API will no longer be possible. All implementations should hence be converted to the exclusive use of released API in the course of FirstSpirit Version 5.0.