FirstSpirit™
Your Content Integration Platform

FirstSpirit™ Release Notes

FirstSpirit™ Version 4.2 Release 4

Version	**1.56**
Status	**RELEASED**
Date	**2011-05-04**
Department	FS-Core
Author / Authors	B.Gutknecht
Copyright	2011 e-Spirit AG
File name	RELN424EN_FirstSpirit_Releasenotes

e-Spirit AG

Barcelonaweg 14
44269 Dortmund | Germany

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

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

e-Spirit AG

Table of contents

1 Introduction

Following Version 4.2 Release 2, FirstSpirit™ Version 4.2 Release 4 (FirstSpirit V4.2R4 for short) is the second release version to be published after the initial release of FirstSpirit Version 4.2 (for further information on the release management, see also *FirstSpirit Release Notes* to version 4.2R2).

The new functions realised in FirstSpirit V4.2R4 are introduced in this document. It is assumed that the reader is already familiar with FirstSpirit™. Wherever possible, the chapters are allocated to the various user groups and the new functions are described from the point of view of the respective group. To understand Chapter 5 to 7 in particular, in-depth knowledge of the respective field is required (template development, administration). For changes concerning the features described in this document and common bug fixes see also the Release Notes which are published with newly released FirstSpirit build versions.

The new "FirstSpirit AppCenter", an infrastructure with which third party software can be integrated in the editorial interface of the FirstSpirit JavaClient, plays an important role in the new version. Options and examples are described in great detail in Chapter 3, from page 16.

2 System Requirements and Installation

While the installation procedure for FirstSpirit Version 4.2 Release 4 had basically not changed compared to V4.2R2, several changes have resulted regarding the system requirements in V4.2R2.

2.1 System requirements

The objective in Version 4.2R4 was to update the web browsers, operating systems and databases supported by FirstSpirit. The products and versions that are maintained or for which maintenance is no longer provided are described in the following chapters.

Several versions have been removed from the maintenance or – in other words – the maintenance of several versions will expire in FirstSpirit 4.2R4. This means that from FirstSpirit 4.2R4, we no longer test whether these versions still function correctly with the new features implemented in FirstSpirit and users are no longer entitled to debugging for these versions. All FirstSpirit Versions 4.x up to and including version 4.2R2 can however, in principle, run with these versions removed from the maintenance. We will also try to enable new functions in version 4.2R4 to run with the configurations removed from the maintenance, but there is no guarantee of this. The maintenance of these versions will expire completely with version 5.0.

For further information on the system requirements for FirstSpirit Version 4.2R4, see the current technical data sheet.

! *For system requirements concerning the FirstSpirit AppCenter see Chapter 3 from page 16.*

2.1.1 Web browsers (for use of FirstSpirit WebEdit)

The following web browsers have been included in the maintenance or maintenance is no longer provided for work with FirstSpirit WebEdit from FirstSpirit Version 4.2R4:

- **Mozilla Firefox:** Versions 3.5 and 3.6 have been included in the maintenance. Support for Version 2.0, on the other hand, expires.
- **Microsoft Internet Explorer:** Version 8 has been included in the maintenance; Version 7 continues to be supported. Maintenance for Version 6, on the other

hand, expires.

- **Other browsers**, e.g. Safari or Google Chrome, are not officially supported.

For information on the use of Mozilla Firefox and Microsoft Internet Explorer for the Integrated preview in FirstSpirit JavaClient, see Chapter 3.1.3 page 20.

2.1.2 Web and servlet engine / application server

A servlet engine which implements the servlet API in version 2.4 is required for the use of FirstSpirit web applications.

The web applications of the FirstSpirit modules fit principally to the version 2.4, but there may be discrepancies, depending on the specific module.

FirstSpirit in version 4.2R4 supports interaction with the following products:

- Apache 2
- Microsoft Internet Information Server (IIS) 6 / 7

2.1.3 Databases

The following databases have been included in the maintenance with FirstSpirit Version 4.2R4, or maintenance is no longer provided:

- **Apache Derby:** The FirstSpirit internal database has been updated to Version 10.5.3.0. This Apache Derby database is, however, not suitable for productive use and should therefore be used for tests only.

Important: Due to the updating of the Apache Derby database, it is not possible for projects which use this database and which were created using FirstSpirit V4.2 Release 4 to be used with older FirstSpirit Versions (see also Chapter 2.2.2 page 14).

- **MySQL:** Maintenance for MySQL Version 4.1 has expired; instead Version 5.1 (both Linux and Windows) has been included in the maintenance. Maintenance continues to be provided for Version 5.0.

- **Microsoft SQL Server:** Maintenance for the Microsoft SQL Server 2000 (Windows) has expired; instead, Microsoft SQL Server 2008 (Windows) has been included in the maintenance. Maintenance continues to be provided for Version 2005 (Windows).

> *There is no separate database layer for the Microsoft SQL Server 2008; the layer for Version 2005 must be selected instead.*

- **Oracle:** Maintenance for Oracle databases Version 9 (Solaris and Linux) has expired, in exchange, Version 11 (Solaris, Linux, Windows) has been included in the maintenance. Maintenance will continue to be provided for Version 10 (Solaris, Linux, Windows).

> *For Oracle, the JDBC driver of the series 10.1 (ojdbc14_10.1.0.x.jar) should be used because problems can arise if the data type `LONG` is used with version 10.2 from 4000 characters on and UTF-8 coding. As an alternative, the compatibility mode for Oracle 9 `LONG` must be activated when using the driver 10.2, because `LONG` is deprecated since Oracle 9. For this purpose the parameter*
>
> ```
> jdbc.property.oracle.jdbc.RetainV9LongBindBehavior=true
> ```
>
> *must be added in the database configuration.*

- **IBM DB2** Maintenance for IBM DB2 Version 8.2 (Linux) has expired, instead, Versions 9.5 and 9.7 (Linux, Windows) have been included in the maintenance. Maintenance continues to be provided for Version 9.1 (Linux, AIX, Windows) and 8.2 (AIX).
- **PostgreSQL:** PostgreSQL Versions 8.0 and 8.2 continue to be maintained, in addition, Versions 8.3 and 8.4 have been included in the maintenance (all Windows and Linux).

	⊕ new in 4.2R4	○ retained	⊖ maintenance expired
MySQL	5.1	5.0	4.1
Microsoft SQL	2008	2005	2000

	⊕ new in 4.2R4	○ retained	⊖ maintenance expired
Oracle	11	10	9
IBM DB 2	9.7 9.5	9.1 8.2 (AIX)	8.2 (Linux)
PostgreSQL	8.4 8.3	8.2 8.0	

> *From Version 4.2R4, the integration of JDBC driver files as a FirstSpirit module is recommended. They should no longer be filed in the* `.../shared/lib` *directory, among other things, to enable various versions of a driver to be operated in parallel. See also Chapter 6.1 page 111.*

2.1.4 Operating systems

The following operating systems have been included in the maintenance with FirstSpirit Version 4.2R4 or maintenance is no longer provided:

2.1.4.1 FirstSpirit JavaClient

- **Microsoft Windows:** Microsoft Windows 7 has been included in the maintenance, support for Microsoft Windows XP, on the other hand, expires. Maintenance continues to be provided for Microsoft Vista.
  The FirstSpirit JavaClient can be operated on both 32 bit and 64 bit platforms. However, as native 32 bit applications are still required for use of the Integrated preview, increased administrative work may have to be anticipated for operation in 64 bit environments.
- **Mac OS X / Linux:** From Version 4.2R4, Mac OS X (Version 10.6) and Linux are officially supported, however, with restrictions: 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 & 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.
If non-Latin characters in FirstSpirit JavaClient are be used under Linux (i.e., e.g. Chinese characters), the Java package (JRE) with Sun-JVM of the respective Linux distribution must be used, e.g. the packages "sun-java6-jre", "sun-java6-plugin" or "sun-java6-fonts" from Debian/GNU and Ubuntu or "java-1_6_0-sun-1.6.0" and "java-1_6_0-sun-plugin-1.6.0" from SUSE and Red Hat. Operation of the JavaClient under other Java systems such as OpenJDK, GNU Java or Apache Harmony is not supported.

! *For further information on special system requirements for use of the **Integrated preview** and of the FirstSpirit AppCenter, see Chapter 3 Page 16.*

! *If Microsoft Windows 7 or Vista is used, the NTLM login module, via which the NTLM authentication is made at the FirstSpirit Server, can only be used if the Windows 7 security settings are lowered. Login via Kerberos ticket (integrated Windows login) is possible in combination with Mozilla Firefox and Microsoft Internet Explorer from V4.2R2 and in this context is the preferred variant. For notes about the login via Kerberos ticket in combination with the FirstSpirit module "Personalisation" see Chapter 7.2.1page 156.*

2.1.4.2 FirstSpirit Server

- **Microsoft Windows:** Microsoft Windows 2008 Server, Microsoft Windows 2008 Server R2 and Microsoft Windows 2003 R2 have been included in the maintenance, support for Microsoft Windows 2000 and Microsoft Windows 2003, on the other hand, will expire.
- **Debian GNU/Linux:** Debian GNU/Linux Version 5.0 will continue to be maintained; maintenance for Version 4.0, on the other hand, will expire.
- **Suse Linux:** Suse Linux Enterprise Server Version 11 has been included in the maintenance; maintenance continues to be provided for Version 10, maintenance for Version 9, on the other hand, will expire.
- **Red Hat:** Red Hat Enterprise Linux Version 5 will continue to be maintained, on the other hand, maintenance for Version 4 will expire.
- **Solaris:** Solaris Versions 9 and 10 continues to be maintained.

- **IBM AIX:** IBM AIX Version 6.1 has been included in the maintenance; Version 5.3 of IBM AIX continues to be maintained.

> ! *If the FirstSpirit Server is operated under **Microsoft Windows 7 or 2008 R2**, problems can occur ("DbChecksumException") if a Berkley database in a version smaller than 3.3.92 is used. Up to FirstSpirit Version 4.2R2, Version 3.3.87 was used, however, Windows 2008 R2 was not officially released there. FirstSpirit Version 4.2R4 contains the Berkley database with Version 3.3.96, so that 4.2R4 is compatible with Windows 7 and Windows 2008 R2.*
> *Microsoft Windows 2008 (without R2) and all preceding Windows versions and use of Windows 7 or Windows 2008 R2 for the FirstSpirit Clients are NOT affected.*

2.1.5 Java

2.1.5.1 Oracle (Sun) JDK

With Version 4.2R4, support for Oracle (formerly Sun) JDK 1.5 will be completely discontinued.

All release tests for FirstSpirit Version 4.2R4 have been performed with the JDK Version 1.6.0_23, not only for the 64-bit version of the FirstSpirit server but also for the 32-bit Version of the FirstSpirit JavaClient. Therefore, use of this version will be released and recommended for 4.2R4.

However, when using JDK 1.6.0_23, the menu bar of the JavaClient can be incorrectly displayed in several interim versions of 4.2R4 as well as in 4.2R2 JDK 1.6.0_23: The menus or menu items are "compacted" and do not have any space between them. This unsightly hitch is corrected with Version 4.2.428, it is not planned to retrospectively apply this to earlier versions.

2.1.5.2 Apple JDK

The Apple JDK, included in the respective currently used Mac OS, is released for FirstSpirit JavaClient. The Apple JDK is not supported for the FirstSpirit Server.

2.1.5.3 IBM JDK

While FirstSpirit Version 4.2R2 under AIX is incompatible with IBM JDK Version 1.6, the IBM JDK 1.6 for is AIX is released for FirstSpirit V4.2R4. At least Version 1.6.0 SR8 must be used as, due to an error in the JDK, IBM JDKs 1.6.0 SR6 and SR7 cause 100 % CPU utilisation even when idle. Maintenance for IBM JDK Version 1.5 under AIX, on the other hand, has expired.

> ! *IBM JDK 1.6 is incompatible with FirstSpirit up to and including Version 4.2R2 (both under AIX and under Linux)!*

2.2 Installation and update

For information on the installation of V4.2R4, see *FirstSpirit Installation Instructions Version 4.2, 4.2R2 and 4.2R4*.

2.2.1 Update

An update from FirstSpirit Version 4.2 or 4.2R2 to FirstSpirit Version 4.2 Release 4 can be easily made by replacing the JAR file (`fs-server.jar`, see *FirstSpirit Installation Instructions Version 4.2, 4.2R2 and 4.2R4*, Chapter 4.9). The following changes have to be made on updating FirstSpirit Servers:

On updating from 4.2R2 to 4.2R4, all installed **FirstSpirit modules** and all **web applications** must also be updated (see *FirstSpirit Manual for Administrators*), as a new signing key is used in 4.2R4 and otherwise error messages can occur.

> ! *The button "update" in the application for Server configuration (area "Modules") has been removed in 4.2R4. Updating can now be effected by using the button "Install".*

For the use of **JDBC drivers** in the database context, for an upgrade to Version 4.2R4 it is advisable to integrate the driver files as a FirstSpirit module, instead of filing them in the directory `.../shared/lib`. See Chapter 6.1 page 111.

The input component **FS_LIST** has been extended with Version 4.2R4. Already existing input components of the type CMS_INPUT_CONTENTAREALIST, CMS_INPUT_CONTENTLIST, CMS_INPUT_LINKLIST, CMS_INPUT_SECTIONLIST and CMS_INPUT_TABLIST can be migrated by adjusting the forms according to FS_LIST. FS_LIST input components which have been created with a FirstSpirit Version before 4.2R4 must also be switched to the new FS_LIST syntax (see Chapter 5.3 page 86).

❗ *If data has been saved in FS_LIST, they can **not** be read and operated by CMS_INPUT_CONTENTAREALIST etc. because of diverging data format.*

From Version 4.2R4, client applications (e.g. the Mozilla engine for using the Integrated preview) will be rolled out by default in an individual directory on the workstation, depending on the used FirstSpirit Version, for 4.2R4 e.g. `\.firstspirit_4.2R4`. The directory used until 4.2R2 `\.firstspirit` will not be deleted by the system, but it must be deleted manually (e.g. to free memory). See also Chapter 8.2, entry "Rollout directory for client applications".

In addition, **incompatible changes to data formats** have been made in FirstSpirit. Therefore, on upgrading to 4.2R4, the following conversion steps are performed automatically:

- **Conversion of started workflows / tasks:** A more efficient data format is used for workflows in Version 4.2R4. Existing and started workflows in a project are automatically converted on upgrading to 4.2R4. This takes place with the first start of the FirstSpirit Server updated to Version 4.2R4. All projects on the server are opened consecutively and the started workflows are individually converted. This process can take some time to complete. The server cannot be used until **after** completion of the conversion.
- **Recalculation and conversion of references:** A more efficient data format is also used in Version 4.2R4 for references. Existing references are automatically converted and recalculated on upgrading to 4.2R4. This is done the first time projects which are on a FirstSpirit Server updated to Version 4.2R4 are opened. Unlike the workflow conversion, the reference recalculation takes place asynchronously. I.e., servers and projects can already be used while the reference calculation is still running. The conversion of the references' data format takes place during the reference recalculation.

2.2.2 Downgrades

Due to the updating of the internal FirstSpirit **Apache Derby database** (see Chapter 2.1.3 page 8), it is not possible to use projects which use this database (not released for productive use) and were created with FirstSpirit Version 4.2R4, with an older FirstSpirit Version.

Due to the new **data formats for workflows and references** introduced in Version 4.2R4 and described in Chapter 2.2.1, a downgrade from 4.2R4 is only possible to 4.2R2 from Version 4.2.219, as the new data formats can only be interpreted from this software status. This also concerns projects which are created with a FirstSpirit Server Version 4.2R4 and are then used on a FirstSpirit Server Version 4.2R2 by means of export/import. Projects which use a Derby database and were created in 4.2R4 can never be used with a 4.2R2 server.

> ! *A downgrade (or even project exports) to older versions than FirstSpirit 4.2 Release 2 is NOT possible!*

Also, in case of a downgrade to 4.2R2, all installed **FirstSpirit modules** and all **web applications** must also be used in the respective version (see *FirstSpirit Manual for Administrators*), as a new signing key is used in 4.2R4 and otherwise error messages can occur.

In addition, **projects with images**, which were created with Version 4.2R4, cannot be used on servers with Version less than 4.2.219. On servers with a version below 4.2.219, images are not displayed in the Media Store due to a missing internal attribute.

In 4.2R4, several **new options** have been added, e.g. in the "View" menu of JavaClient (see Chapter 4.2.2.3 page 37). If a project created with Version 4.2R4 is used, e.g. on a FirstSpirit Server Version 4.2R2, warnings appear when the server is started for the first time, e.g.:

```
WARN  24.06.2010 12:22:08.063
(de.espirit.firstspirit.common.xml.XmlSerializer): no field
de.espirit.firstspirit.server.projectmanagement.InlinePreviewConst
raintsDTO#_mediaPlayerFileExtensions
```

If changes are made to the Project properties, the next time the server is started no more warnings are issued.

3 The FirstSpirit AppCenter

e-Spirit sees itself as a product manufacturer, who consistently pursues a best-of-breed strategy. This includes systematic focussing of all in-house implementations on the core components of the FirstSpirit product and deliberate transfer of specific functions in third party products of the respective market leaders. The decisive requirement for successfully implementing this popular outsourcing idea in a software product is "seamless integration": there must be no break between the products used for the end user! The user prompting must appear fully integrated, seamless and visually forming a unified whole to the user.

This idea of the seamless integration of third party applications into the FirstSpirit editing environment is called "AppCenter". The FirstSpirit AppCenter provides a defined area within the editing system, in which independent applications can run, which are not part of FirstSpirit (so-called "AppCenter applications").

Examples of AppCenter applications available with FirstSpirit Version 4.2R4 are the integration of Microsoft Office (see Chapter 4.2.2.1 page 34 and Chapter 6.2.3 page 129) or the functions for integrated image editing (see Chapter 4.2.2 page 32 and Chapter 4.2.3 page 40). The Mozilla Firefox and Microsoft Internet Explorer web browsers integrated in Versions 4.2R0 and R2 are also AppCenter applications, they are called "Integrated preview". e-Spirit has implemented all these AppCenter applications as product components. There is also a range of AppCenter applications, which are implemented as FirstSpirit modules. These AppCenter modules can be developed not only by e-Spirit itself, but also by a partner.

The following examples, which have already been successfully integrated by e-Spirit as part of the AppCenter, give an impression of the possibilities the AppCenter offers beyond the applications currently implemented as product components: By integrating Google Maps or Google Earth, geocoordinates can be simply and intuitively used in the FirstSpirit editing system, integration of the online video offer of MovingImage24 enables videos to be selected and integrated into FirstSpirit with the click of a mouse, and with integration of the Fotolia and Pixelio online photo data bases, it is possible to directly search for photo material in the FirstSpirit editing system and to copy the required photos into FirstSpirit, also with the click of a mouse.

In technical terms, AppCenter consists of a number of interfaces, which have been released by e-Spirit for use by partners, so that they can implement or integrate specific applications as part of the AppCenter (see also Chapter 3.5 page 24). The available AppCenter interfaces are currently limited to the infrastructure required for the integration of web applications. Appropriate interfaces for the integration of native applications (analogous to Microsoft Office, see also Chapter 3.1 page 17) or Java applications (analogous to the integrated Java image editing, Java Image Editor, see also Chapter 3.2 page 21) have already been implemented but are not yet publicly available. The general term used to describe the implementation of AppCenter applications and their integration in FirstSpirit is "application integration".

Requirements for and limitations to the use of AppCenter applications are described in the following, and reference is made to further documentation.

3.1 Native application integrations in 4.2R4

The AppCenter integrations already implemented as part of the FirstSpirit product with Version 4.2R4 offer editors a high degree of added value for their editorial work: Integration of the Mozilla Firefox and Microsoft Internet Explorer web browsers not only allows direct previewing in the Client, but also visualises the relationship between the content entered in the Client and its effect or display on the website and therefore makes it easier to navigate through the project. The plug-ins officially available for these browsers can also be used to display PDF, Flash files, etc. in the integrated preview. The integrated Microsoft Office and OpenOffice applications can be used to edit and save Microsoft Office Word, Excel and PowerPoint or OpenOffice Writer, Calc and Impress files directly in the JavaClient. OpenOffice integration is currently in the BETA test phase and has not been officially released.

To use these native integrations, the respective application must be installed on the workstation. The integrations in JavaClient fall back on the local configuration of the applications. One exception is Mozilla Firefox: this is delivered with the FirstSpirit core product and is managed by FirstSpirit itself.

When using the integrations named above in the Client, limitations can on principle occur, e.g. because several of the integrated applications do not fully work together with all platforms or bit sizes (32 or 64-bit, see Chapter 3.1.1 page 17). For example, under Linux, Microsoft Internet Explorer cannot be used as the web browser, and Microsoft Office cannot be used as the office application.

3.1.1 Requirements regarding 32/64-bit architectures

The least interoperability problems currently occur if **32-bit systems** are used, as the software infrastructure used (operating system, JDK and native application) all exist in 32 bit. Due to the known limitations of the 32-bit architecture, especially with regard to the main memory (RAM), the introduction of 64-bit systems is progressing rapidly – with Microsoft Windows 7 also on the desktop. Within the scope of native application integration, on principle, this gives rise to a new level of complexity, which is briefly outlined in the following:

For compatibility reasons, current **64-bit operating systems** such as Windows 7 or Linux are basically also able to run 32-bit applications. This produces the problem that both 64 and 32-bit applications can exist on a 64-bit operating system. This specifically also applies to the JDK/JRE: this can exist in a 32 and/or a 64-bit version on a 64-bit operating system.

The FirstSpirit JavaClient itself is compatible with both a 32 and with a 64-bit Java. The difference lies in the application integration of the JavaClient: Also, on a 64-bit operating system (on principle) on 32-bit applications can be integrated in conjunction with a 32-bit Java!

But use of a 64-bit Java system is principally desirable on a 64-bit operating system. In FirstSpirit JavaClient, the application integration is realised so that **all** integrated native applications are also integrated as a 64-bit variant, i.e. no 32-bit applications are started from a 64-bit Java! (Error message: "Can't load IA 32-bit .dll on a AMD 64-bit platform")

Therefore, all applications to be integrated must exist as a 64-bit version, e.g. "Microsoft Internet Explorer (64-bit)". Unfortunately, several applications (especially under Microsoft Windows) are currently not yet available as a 64-bit variant (e.g. Mozilla Firefox and OpenOffice for Windows 64-bit). Therefore, these applications cannot be used in conjunction with a 64-bit Java. It is only possible to use these applications if a 32-bit Java is used on the 64-bit operating system!

If using 64-bit web browsers, e.g. Microsoft Internet Explorer, all the plug-ins used in the browser must also exist as a 64-bit variant. The Adobe Acrobat Reader plug-in, for displaying PDFs, for example, is currently not available as a 64-bit variant. "PDFexchange" can be used as an alternative.

With FirstSpirit Version 4.2R4, general 64-bit operation will also be possible in conjunction with application integration, not only for Windows but also for Linux! However, due to the inherent complexity, 64-bit support for application integration is still experimental, as all possible combinations (operating system, Java, third-party application) have not yet been fully tested for compatibility. It is planned to release 64-bit support during FirstSpirit Version 4.2R4. As soon as it has been released, it will accordingly also be included in the FirstSpirit *Technical Data Sheet*. Therefore, there may currently still be discrepancies between these *Release Notes* and the current *Technical Data Sheet*.

It should also be noted that the available combination options are limited due to the described dependencies (see table):

	Mozilla Firefox	Internet Explorer	Preview for PDF, SWF etc.	Microsoft Office Applications	OpenOffice Applications (**BETA**)
Windows 32 bit	⊕	⊕	⊕	⊕	⊕
Windows 64 bit	⊘ only 32 bit	⊕ 32 bit and 64 bit (**BETA**)	⊘ partially only 32 bit	⊕	⊘ only 32 bit
Mac OS 10.6	⊕ only 32 bit	⊖	⊕ only 32 Bit	⊖	⊖
Linux 32 bit	⊕	⊖	⊕	⊖	⊕
Linux 64 bit	⊕ 32 bit and 64 bit (**BETA**)	⊖	⊘ partially only 32 bit	⊖	⊕ 32 bit and 64 bit (**BETA**)

Legend: ⊕ compatible ⊘ with exceptions ⊖ incompatible

The indication of bits refers in each case to the application **and** the JRE!

3.1.2 Use of AppCenter applications

With regard to use of AppCenter applications (Chapter 3.1 up to and including 3.4), it must be noted that, in principle, FirstSpirit provides the interfaces required for the application integration, but in general it does not have any influence on the integrated applications themselves. Integrated third party applications are not a FirstSpirit product component. Among other things, this means that the responsibility for the functionality of the integrated applications lies with the manufacturer or the customer or partner who implements the application.

Problems can be reported within the scope of the FirstSpirit product support and, (where possible) will be corrected if they lie on the level of the integration interface. However, **no** claims can be made against e-Spirit for debugging **within** the third-party applications described in Chapters 3.1 to 3.4.

3.1.3 Web Browser Integration

Depending on the settings in JavaClient or in the project settings (project administrator), Microsoft Internet Explorer and Mozilla Firefox are used for display of the HTML preview as well as for media from the FirstSpirit Media Store of the type PDF and Adobe Flash.

Both browsers are currently started as 32-bit applications. A special version of Firefox integrated in JavaClient is used for use of Mozilla Firefox so that it is not necessary to install it locally on the workstation. If Microsoft Internet Explorer is used, an installation of the 32-bit variant of this browser must be available on the workstation. In addition, at present, the JRE must always be a 32-bit application in order for the integrated preview to function. Support for 64-bit JREs in conjunction with the integrated preview is planned, but requires a stable 64-bit Mozilla core, which is not yet available under Windows.

The browsers are used to display special file formats, e.g. PDF, SWF, etc. Support for these file formats is platform-independent, and is therefore, in principle, also available under Linux and Mac OS, though always in the 32-bit variant only. However, it is not possible to guarantee that all browser plug-ins correctly function with the integrated preview operating system-wide. In our experience, plug-ins of known manufacturers, such as Adobe Flash and Adobe Acrobat Reader, do not cause problems.

The integrated preview will very probably not be available **under Mac OS** in the initial release version of 4.2R4, or if it is, then only to a very limited extent. Work is being done on correcting the error, however, as it is a matter of technical problems with a third party component, it is not possible to ensure debugging will be achieved in time for the release date. In addition, for technical reasons, the function for the display of PDF documents in the integrated preview is currently not available on Mac OS X.

! *It is recommended to use an Internet Explorer, version 8 and higher. Internet Explorer up to version 8 do not support any Base64 decoding. This can lead to problems with injecting picture elements within the scope of the application integration (e.g. when displaying the input component FS_BUTTON (see Chapter 5.4 page 99) in the Integrated preview or when integrating image databases, see also Chapter 3.4 page 23).*

! *The display of Java applets is not supported by the JxBrowser, which is used for the integrated preview ("View" menu / "Browser Engine").*

3.1.4 Microsoft Office Integration

Microsoft Office file formats can only be displayed under Windows and this requires a local Office installation. The appropriate application for the format of the selected medium is used, i.e. for example, Microsoft Word for media of the type DOC, Microsoft Excel for media of the type XLS, etc. In principle, a 64-bit JRE can also be used; however, the integrated preview is then not available.

! *Using the Integrated Preview for media with Microsoft Office and externally as Microsoft Office desktop application (e.g. started via Windows or in the JavaClient) at the same time, can lead to problems under certain circumstances and is no guaranteed feature of FirstSpirit integration. In this case, either the Integrated preview for Office documents in FirstSpirit* ***or*** *the respective Microsoft Office desktop application should be used, no* ***mixed*** *mode.*
For information on other limitations, see also Chapter 4.2.2.2 page 36.

3.1.5 Integration of the Windows Media Player

If FirstSpirit Media Store files with file format AVI, MPG, MPEG, WMV, ASF, MP3, MP4 are to be played using the Windows Media Player integration, it must be installed on the workstation. Use is only possible under Microsoft Windows.

3.2 Java application integrations in 4.2R4 (Java Image Editor)

Unlike Picnik and Pixlr (see Chapter 3.3 page 22), Java Image Editor is an integrated Java application. Like Mozilla Firefox, it is delivered with the FirstSpirit core product and is managed by FirstSpirit itself. It was implemented during the AppCenter development work as an example of the integration of a Java-based third party application and was originally not intended to be part of the product. However, in practice it was found that a range of (simple) image manipulations could be performed with the help of this image editing software, and so this third party implementation was licensed and released for use in FirstSpirit.

Analogous to the other integrations in Chapter 3.1, release for use of the Java Image

Editor is explicitly "without functional guarantee" for the application itself. I.e. e-Spirit does not provide any guarantee whatsoever, neither explicit nor implicit, for the image editing functions, instead the release is for use "as-is" (see also Chapter 3.1.2 page 19). If the use of image editing functions is a critical production function, external image editing software with relevant manufacturer support should be used.

At present, Java Image Editor is only available under Mac OS to a very limited extent. For information on other limitations, see also Chapter 4.2.3.2 page 42.

3.3 Application integrations in the BETA stage

Several integrations delivered with the core product are currently still in the BETA stage and are not released for productive use:

OpenOffice: OpenOffice file formats can be used on Microsoft and Linux operating systems. As with the integration of Microsoft Office applications, the relevant OpenOffice application must be installed on the workstation. When using OpenOffice under Linux, display problems can still occur at present, but in practice the display differs only marginally from the Windows version.

Google Docs: Google Docs can be used to edit office documents in FirstSpirit JavaClient under Mac OS X too. This requires an active connection with the internet as well as a Google account (http://docs.google.com). Locally installed software is therefore no longer necessary for the preparation and editing of office documents – but in return, all documents must be uploaded onto a Google server before they are worked on. This sensitive data protection topic, the current still very large limitations to the document editing (e.g. limitation of the file size and incompatibilities) and the use of a not yet released API mean that Google Docs integration has a prototype character only in FirstSpirit Version 4.2R4 and is **not** released for productive use. In several cases, documents can even be damaged by the editing in Google Docs. This is not a FirstSpirit problem.

The free version of Google Docs only supports maximum file sizes of 1 MB. At present, Google Docs can only be used in JavaClient with the "Internet Explorer" browser engine. For information on other limitations, see also Chapter 4.2.2.2 page 36 and the relevant internet site of the provider.

Picnik and Pixlr: The graphic engines Picnik and Pixlr access the online photo / image editing services www.picnik.de and www.pixlr.com and therefore each require an active connection with the internet. Here, too, the photos/images are in fact uploaded onto the server of the respective provider, where they are also edited. If used, this should be taken into account with regard to data protection. For further information, please also refer to the relevant internet sites of the providers.

3.4 Customised application integrations

The FirstSpirit AppCenter enables customers and partners to integrate individual applications into JavaClient and therefore to adapt the Client to their special needs. The already integrated browsers, in particular, provide infrastructure, with which it is very easy to implement the integration of Online offers, e.g. for fast internet searching for text or image material or for use of Google services. An active internet connection, possibly and accounts, etc. is required for use of online offers.

Even if customised AppCenter applications are integrated and used, it must always be noted that FirstSpirit provides the interfaces required for the application integration, but in general it does not have any influence on the integrated applications themselves. Integrated third party applications are not a FirstSpirit product component. Among other things, this means that the responsibility for the functionality of the integrated applications lies with the manufacturer or the customer or partner who implements the application.

Problems can be reported within the scope of the FirstSpirit product support and, (where possible) will be corrected if they lie on the level of the integration interface. However, **no** claims can be made against e-Spirit for debugging **within** the third-party applications described in Chapters 3.1 to 3.4.

! *If using **Mac OS and Linux**, it must be noted that the reliability performance of AppCenter applications cannot be guaranteed under these platforms; therefore, these platforms are not supported for AppCenter applications.*

Among other things, the input component FS_BUTTON was also introduced in 4.2R4 for the integration of the user's own, simple AppCenter applications. This is used to provide an individually definable icon or button, which can be clicked to perform a script or class stored in FirstSpirit (see Chapter 5.4 page 99), which possibly implements an AppCenter application. Several web applications for use in the AppCenter can be installed in the server properties, "Web Applications" area (see Chapter 6.3 page 131).

A licence is required for use of the customer's own applications in the AppCenter. For further information please refer to Chapter 6.5.2 page 137.

3.5 Further documentation

Detailed information on the FirstSpirit AppCenter concept, the technical architecture and illustrated application examples are also given in the *FirstSpirit Whitepaper 2011*.

Information on how individual applications can be implemented for FirstSpirit is given in the *FirstSpirit Developer Manual for Components* (only available in German). The accompanying zip archive (MODDEV4x_modexamples.zip) contains examples of all general module and component development in FirstSpirit; further examples for AppCenter applications are available through the e-Spirit helpdesk. Documentation on the FirstSpirit AppCenter is also provided for developers.

In addition, an application integration API ("Application-API") is available. This remains stable in 4.2R4, but it may possibly change with FirstSpirit Version 5. See also *FirstSpirit Online Documentation*, section on "Template development" / "FirstSpirit API" / "Application integration".

4 New/Modified Functions for Editors

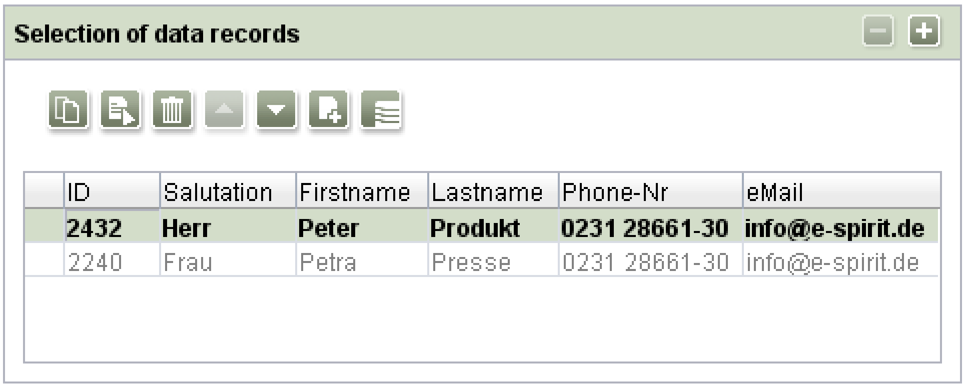
4.1 New/Modified functions in JavaClient and WebClient

4.1.1 Input components: Further development of FS_LIST

Since its introduction in the initial release version of 4.2, the input component FS_LIST has shown the functions of the input component CMS_INPUT_CONTENTAREALIST. The objective of the consolidation of the FirstSpirit input components started in 4.2 should be to group together all input components with a list character in FS_LIST. Therefore, in Version 4.2R4 the functions of the following input components have been added:

- CMS_INPUT_CONTENTLIST
- CMS_INPUT_LINKLIST
- CMS_INPUT_SECTIONLIST
- CMS_INPUT_TABLIST

FS_LIST can also be used in the familiar way for these input components. In some cases other icons are used and the selection dialog familiar from other input components including search function.



Selection of data records

	ID	Salutation	Firstname	Lastname	Phone-Nr	eMail
	2432	**Herr**	**Peter**	**Produkt**	**0231 28661-30**	**info@e-spirit.de**
	2240	Frau	Petra	Presse	0231 28661-30	info@e-spirit.de

Figure 4-1: FS_LIST for the selection of data records

The following icons are available, depending on the defaults set by the template developer and the application purpose of the input component:

Add (only for data records), click this icon to open a selection dialog to select a data record

New, in the case of data records a window opens when clicking this icon to enter a new data record, in the case of sections and links a selection dialog will open for selecting a correspondent template

Copy, in the case of data records a copy of the selected data record will be created in the related data source if using this icon and added to this list.

Remove, click this icon to delete the selected list element from the list

Delete (only for data records), click this icon to delete the selected data record from the list and from the related data source. This data record can not be re-selected later on.

Move up / down, click this icon to move the selected list element one position up or down

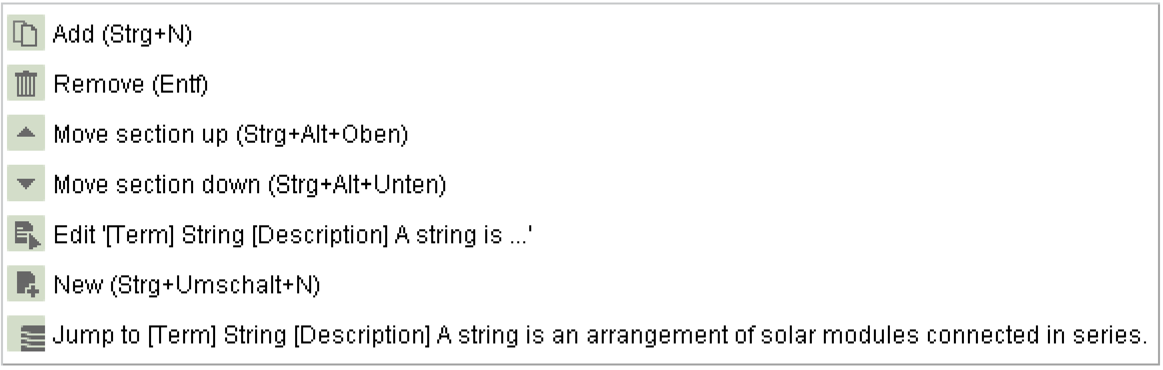
Edit, click this icon to open a window for editing the selected list element. In the View mode, the icon is displayed.

View, if overview and detailed information are shown in the input component, use this icon to open or close the detailed view of the entry which is selected in the overview. If only the overview is shown, use this icon to display the details of the selected entry in a separate window.

Jump to, click this icon to switch to the Content Store, in order to edit the selected data record there.

Open in separate window, use this icon to open the selected list entry in a separate window

In addition, it is also possible to edit the lists using a context menu or keyboard shortcut. The same functions are available here as provided by the icons:



Add (Strg+N)

Remove (Entf)

Move section up (Strg+Alt+Oben)

Move section down (Strg+Alt+Unten)

Edit '[Term] String [Description] A string is ...'

New (Strg+Umschalt+N)

Jump to [Term] String [Description] A string is an arrangement of solar modules connected in series.

Figure 4-2: FS_LIST – Context menu

All data entered in the project via the input components named above are available unchanged following a changeover to FS_LIST. The changeover is usually carried out by the template developer.

! *In **WebEdit**, FS_LIST in 4.2R4 is only supported in a rudimentary way; at present, only the functions of the input component CMS_INPUT_CONTENTAREALIST (list for managing section contents) are available. More functions can be provided using the so called applet mode. However, this will only be a transition solution; native FS_LIST support will exist from FirstSpirit Version 5.0.*

For information on the migration to FS_LIST and the configuration options, see also Chapter 5.3 page 86.

4.1.2 New input component: FS_BUTTON

Depending on the template developer's settings, the new input component FS_BUTTON represents a button

Figure 4-3: FS_BUTTON – button

an icon

Figure 4-4: FS_BUTTON – icon

or a simple link

FS_BUTTON

Figure 4-5: FS_BUTTON – link

When clicked a function is performed, which can be defined by the template developer. Depending on the configuration of the input component, the function can also be performed when an object (from FirstSpirit, files from the local workstation, etc.) is dragged onto this control using the mouse.

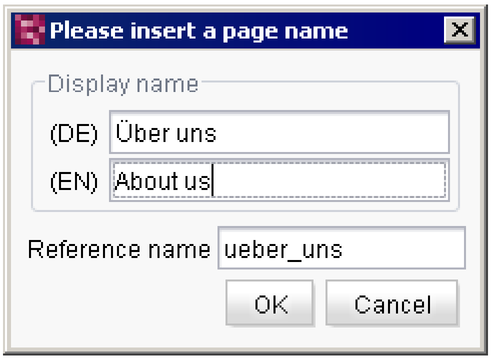
The input component FS_BUTTON is currently not supported in FirstSpirit WebClient. Appropriate support is planned for FirstSpirit Version 5.

4.1.3 Reference names: Automatic conversion of special characters

Reference names may not contain special characters in any store, in FirstSpirit JavaClient; upper case (capital) letters must also not be used for reference names. In JavaClient, these invalid characters are removed during the input, upper case letters are converted to lower case letters. In FirstSpirit WebClient, special characters are automatically converted into underscores (_) and upper case letters are also allowed.

Depending on the defaults set by the administrator, from FirstSpirit Version 4.2R4, special characters in reference names are automatically transformed into valid characters, in JavaClient, directly on entry on creating a FirstSpirit object or on changing a reference name (context menu "Extras" / "Change reference name"), in WebClient after saving. For this, the administrator defines rules about which special characters are to be converted into which valid characters. A "Ü" could then, e.g. automatically be transformed into a "ue", spaces into underscores, etc. This is useful, e.g. if reference names are used for the generation of URLs on generating a project.

If the reference name in **JavaClient** is generated from the display name (e.g. on creating pages in the Page Store), the reference name is also transformed accordingly. E.g. the display name "Über uns" (About us) could become the reference name "ueber_uns" on input:



Please insert a page name

Display name

(DE) Über uns

(EN) About us

Reference name ueber_uns

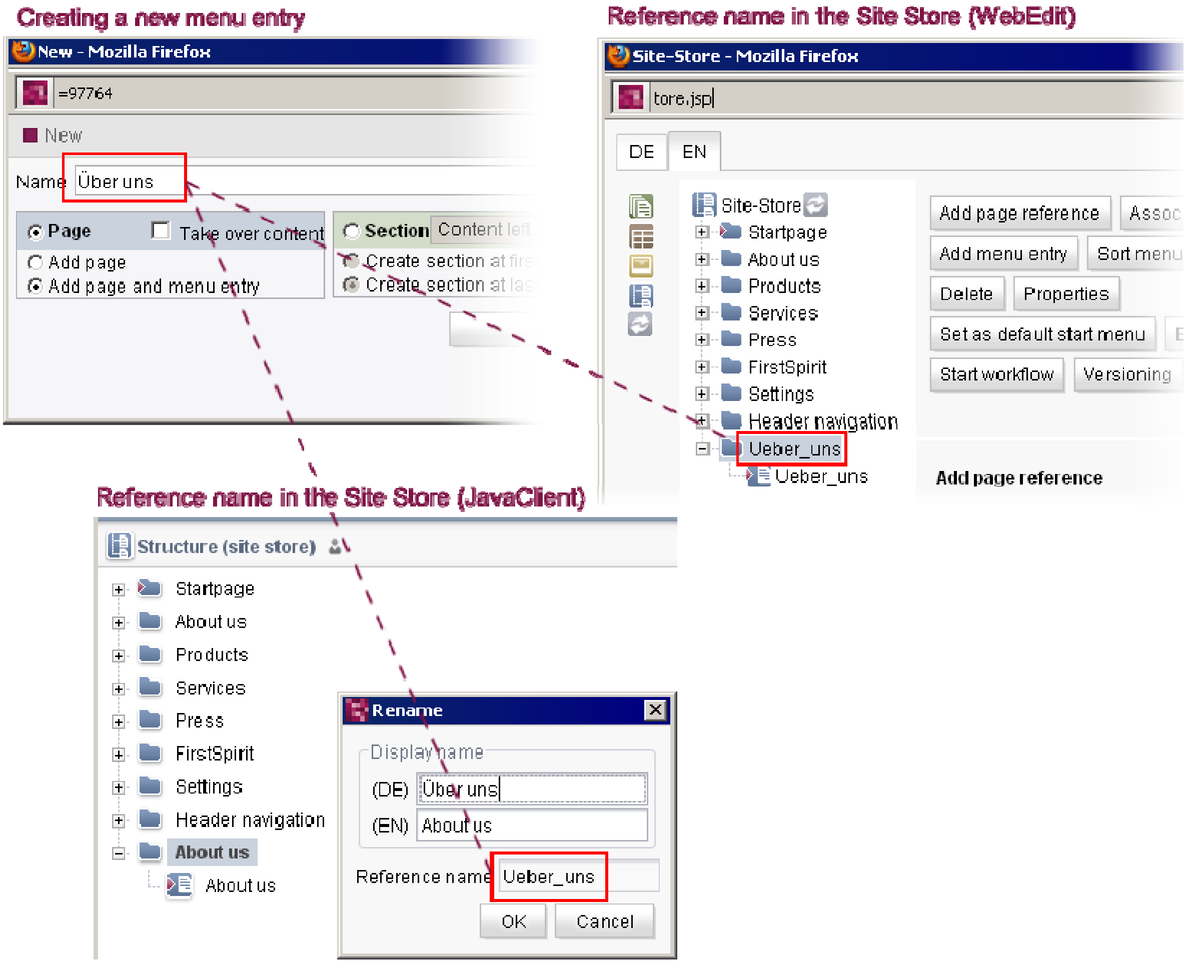
Figure 4-6: Reference name on creating a new page

! *Reference names are only displayed and can only be changed if the administrator has set this for the editor or project.*

! *In general (regardless of the transformation rules given by the administrator), in JavaClient upper case letters in reference names are transformed into lower case characters directly on input. Special characters for which the administrator has not given a replacement rule cannot be entered in the "Reference name" field.*

! *Depending on the configuration by the administrator, the replacement result in the "Display name" field can differ from that in the "Reference name" field.*

If display names from which reference names are formed are given with special characters in **WebEdit**, they are also transformed accordingly after saving, e.g.



Name: Über uns

● Page ☐ Take over content ○ Section

○ Add page
● Add page and menu entry

Display name
(DE) Über uns
(EN) About us

Reference name: Ueber_uns

Figure 4-7: Transforming reference names in WebEdit

4.2 New/changed functions in JavaClient

4.2.1 New icons for files of the Media Store

Whereas different icons for pictures and files, language-dependent and language-independent, have been used in the tree structure of the Media Store so far, now also specific icons, geared to the file type, are displayed for files. In this way, you can read directly from the tree structure of which type a file in the Media Store is, without being obliged to switch to the detailed view of the file. These icons are used in the folders' overview of the Media Store (see Figure 4-8), on tabs of the working spaces, in the selection dialogs etc., e.g.:

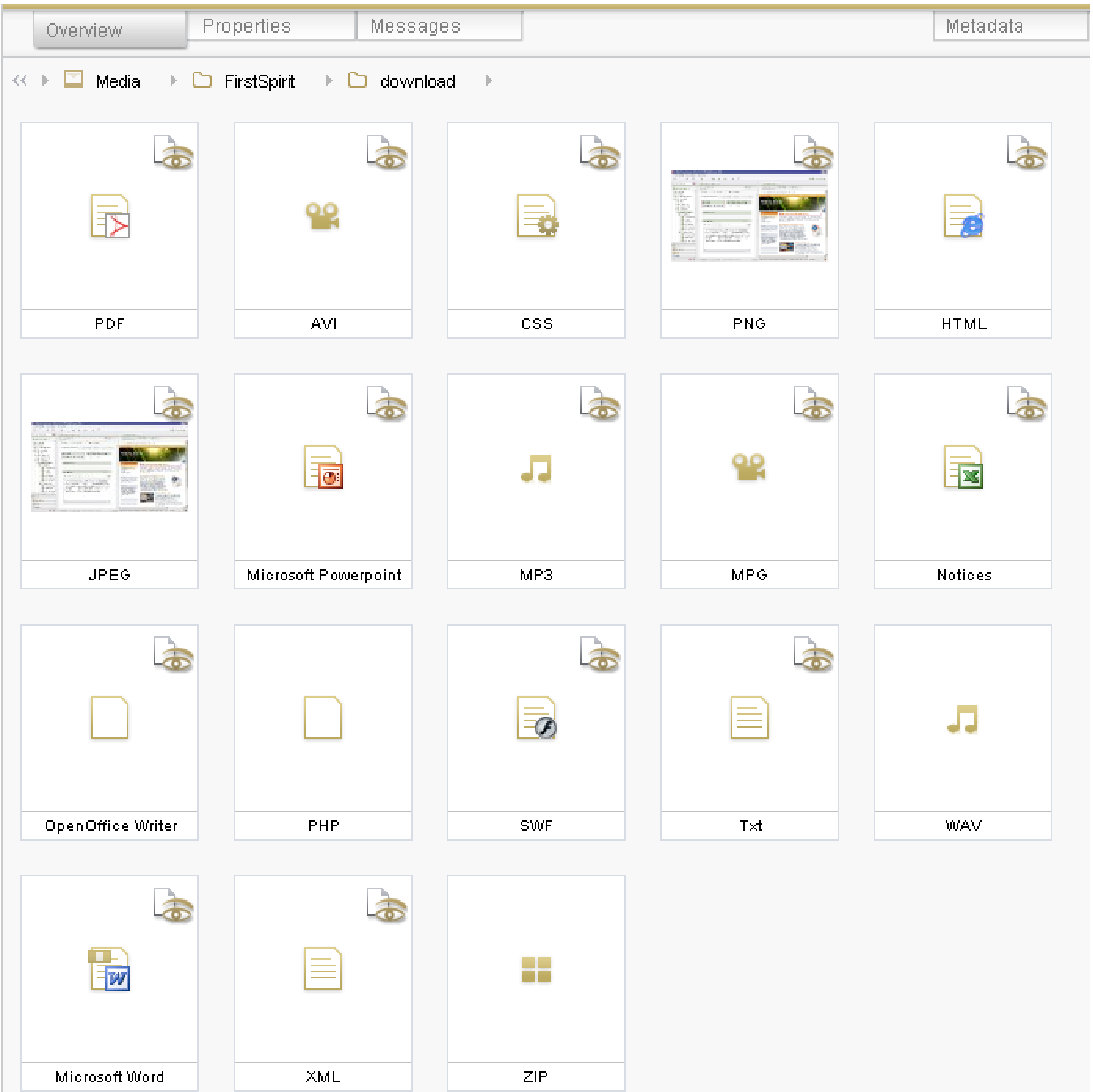


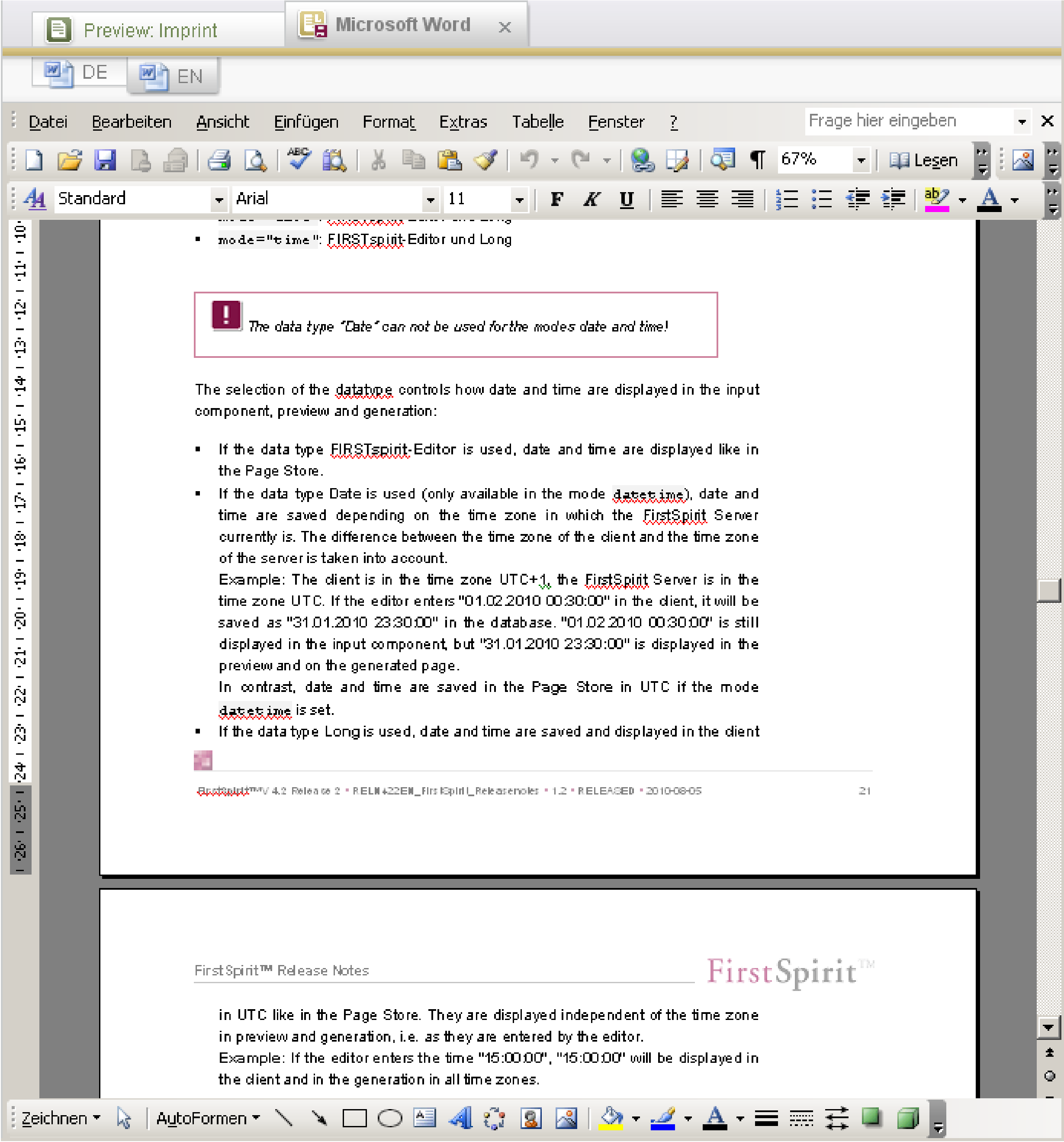
Figure 4-8: Media Store – folders overview

File types for which no separate icon is used are displayed with the 🗋 icon, language dependent files are especially identified:↗🗋

If the Integrated preview is enabled for media (see Chapter 4.2.2.3 page 37), the 👁 icon indicates that the file concerned can be displayed in the Integrated preview (see Chapter 4.2.2 page 32). The medium will open in the integrated preview by one click on this icon.

4.2.2 Functional enhancements of the Integrated preview

From 4.2R4, the "Integrated preview" introduced in JavaClient in the initial version of FirstSpirit 4.2 can not only be used for a preview of the currently edited content as to date, but also (if correctly configured (see Chapter 4.2.2.3 page 37)) other file formats (e.g. Microsoft Word or Excel files, see Chapter 4.2.2.1 page 34) and the FirstSpirit Online Documentation (see Chapter 4.2.2.4 page 39) can now also be displayed there, namely with the help of an application matching the format.



- mode="time": FIRSTspirit-Editor und Long

! *The data type "Date" can not be used for the modes date and time!*

The selection of the datatype controls how date and time are displayed in the input component, preview and generation:

- If the data type FIRSTspirit-Editor is used, date and time are displayed like in the Page Store.
- If the data type Date is used (only available in the mode datetime), date and time are saved depending on the time zone in which the FirstSpirit Server currently is. The difference between the time zone of the client and the time zone of the server is taken into account.
  Example: The client is in the time zone UTC+1, the FirstSpirit Server is in the time zone UTC. If the editor enters "01.02.2010 00:30:00" in the client, it will be saved as "31.01.2010 23:30:00" in the database. "01.02.2010 00:30:00" is still displayed in the input component, but "31.01.2010 23:30:00" is displayed in the preview and on the generated page.
  In contrast, date and time are saved in the Page Store in UTC if the mode datetime is set.
- If the data type Long is used, date and time are saved and displayed in the client

FirstSpirit™V 4.2 Release 2 ▪ RELN422EN_FirstSpirit_Releasenotes ▪ 1.2 ▪ RELEASED ▪ 2010-08-05 21

FirstSpirit™ Release Notes

in UTC like in the Page Store. They are displayed independent of the time zone in preview and generation, i.e. as they are entered by the editor.
Example: If the editor enters the time "15:00:00", "15:00:00" will be displayed in the client and in the generation in all time zones.

Figure 4-9: Display of a Microsoft Word document in the Integrated preview

To this end, several **tabs** in the Integrated preview can be simultaneously opened, e.g. using

- the Preview icon of the JavaClient toolbar,
- Ctrl + P,
- the "Preview" entry of the Context menu
- the new Preview icon on the folder level of the Media Store (see also Figure 4-8)
- the switch to Edit mode (Lock)

However, the tabs cannot be moved like in the editing area (middle column of JavaClient): They are always in the order in which they were opened, until they are closed. On the far left-hand side is the tab with the preview of the output channels (if the "Integrated preview for content" is activated, see also Chapter 4.2.2.3 page 37). The tabs show the name of the respective medium (or "Help", if FirstSpirit Online Help is opened) and the icon for the file type. In addition, the icon is used to show whether it is a language-dependent medium or not and whether the medium has changes which have not been saved.

The "Dis-/Enable tree synchronisation" function familiar from the middle editing area of JavaClient is not available in the Media preview: here synchronisation is constantly active. I.e., if you switch to another tab in the Integrated preview, this medium is switched to in the tree structure and is therefore also displayed in the editing area. If a medium has been opened in the preview from a folder overview (see Chapter 4.2.1 page 31), this is displayed by the Folder icon 📁 on the respective tab. If this tab is clicked, the folder is displayed in the editing area instead of the medium it contains. A tooltip shows the name of the folder. Open the context menu on the Folder icon to switch to the folder overview.

Further information on the currently displayed file is displayed on the **hanging tabs**:

- "Medium": This is a language-independent medium, a file type icon is also displayed for several file formats.
- "DE" / "EN", etc.: This is a language-dependent medium, a file type or application icon is also displayed for several file formats, it is possible to switch to the Media preview in the desired language via the respective language tab. With PDFs and Flash files the icon of the respective selected browser is displayed.

Close preview / **Close other previews**: These context menu entries are available on each tab, as in the editing area (middle column of the JavaClient).

×: This icon can be used to close the individual tabs apart from the one containing the preview of the output channels. If the browser is changed ("View" menu / "Browser engine"), close the tabs also.

4.2.2.1 Preview of media

In order to distinguish between the Integrated preview in the Media Store from the Integrated preview for HTML content familiar to date, the HTML preview used in the right-hand column of JavaClient to date is called "Integrated preview for content", the new preview as the "Integrated preview for media". In the Integrated preview for media, depending on the default settings made by the project administrator (see Chapter 6.2 page 125 ff.), media can be displayed as follows:

- **Images** are displayed in the respective language and different resolutions via a java-based application. To display a certain resolution, the desired resolution must be selected in the resolution list and the Preview icon clicked, or alternatively, activated by means of a double-click. According to the specifications of the project administrator an image processing application can be available instead of the preview (see Chapter 4.2.3 page 40).
- **PDF and Flash documents** are displayed with the help of a plug-in of the browser used, with which these documents can be displayed. The relevant plug-in must be installed on the workstation computer of the editor and correctly configured. All the plug-in's functions are available.
- **Office documents** (Microsoft Word, Excel, PowerPoint; OpenOffice Writer, Calc, Impress [BETA test stage only!]) can be opened in the respective Microsoft Office or OpenOffice application [BETA test stage only!] installed on the workstation computer or Google Docs [BETA test stage only!]. In this way, a Word document, e.g. can be edited directly in the FirstSpirit Media Store. To do this, it is necessary to switch to Edit mode. Changes must be saved using the Save functions of JavaClient, so that the changed document is available in JavaClient.

> ! *If changes are made to documents, e.g. Microsoft Excel sheets, which are not in FirstSpirit Edit mode, information messages of the respective Microsoft Office application can be displayed during editing of the documents. These may be closed by using "OK".*

> ! *The OpenOffice and Google Docs integration are currently in the BETA test stage and are not released officially yet!*

- **Audio and video files** (e.g. MP3, WMV, AVI, MPG) are played in the Integrated preview via the Windows Media Player.
- **Text files** (e.g. TXT, HTML, XML, CSS) are displayed via a text editor integrated in FirstSpirit. FirstSpirit syntax is highlighted in colour (syntax highlighting).

! *Correct saving as an image or file in the FirstSpirit Media Store is important so that the desired document from the Media Store (e.g. Word file) can be correctly displayed and processed*

The media file can also be displayed in a separate window ("View" menu / "Display area of the preview" / "in separate window").

4.2.2.2 Notes and restrictions

Microsoft Office applications

Using the Integrated Preview for media with Microsoft Office and externally as Microsoft Office desktop application (e.g. started via Windows or in the JavaClient) at the same time, can lead to problems under certain circumstances and is no guaranteed feature of FirstSpirit integration. In this case, either the Integrated preview for Office documents in FirstSpirit **or** the respective Microsoft Office desktop application should be used, no **mixed** mode.

In the case of Microsoft Excel for example, Excel files can not be edited by means of the desktop application, if an Excel file is displayed in the Integrated preview: neither already opened Excel files can be edited, nor can Excel files be opened, for example by double click, in the desktop application. If all tabs of the Integrated preview are closed which contain a preview of an Excel file, Excel files can be opened and edited again.

Some menu functions of the Office applications can not be effected for technical reasons (e.g. "File" / "New"). If for example pictures are linked in documents (but not embedded), access to them is only possible after uploading them in the Media Store if the paths are correct. There may be problems if the paths are relative. If they are absolute, there may be problems if the paths differ for different editors, working on the same project, for example because of individual system configurations (e.g. differing drive letters).

Display of PDF files

During the start of the JavaClient, an Adobe Reader window may possibly open. This occurs for technical reasons and can be prevented by suitable settings in the local installation of the Adobe Acrobat Reader on the workstation. Depending on the version used, this is done by opening the "Preferences" menu item of the "File" or "Edit". Then activate the "Display PDF in browser" or "Web Browser Integration" option there.

Google Docs

Use of Google Docs for editing office documents is not currently an assured product property. In practice, a range of problems are still to be expected. It is also possible that documents are damaged by the editing in Google Docs. This is NOT a FirstSpirit problem!

Use of Google Docs requires signing on in Google; to this end, a corresponding query is displayed in JavaClient:

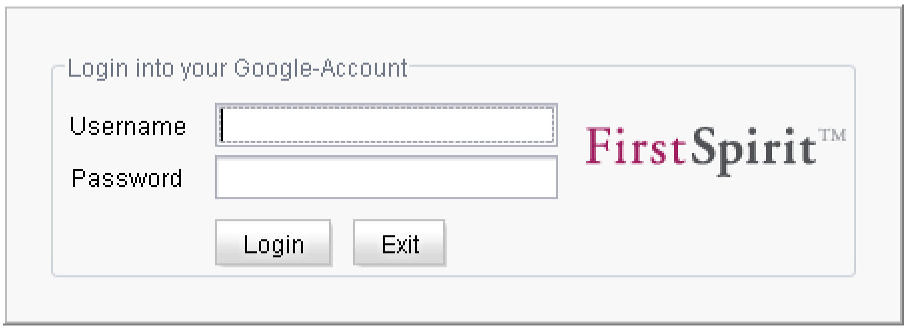


Figure 4-10: Signing on for Google Docs use

The free version of Google Docs only supports maximum file sizes of 1 MB. At present, Google Docs can only be used in JavaClient with the "Internet Explorer" browser engine.

4.2.2.3 Configuration using the "View" menu

The **"View" / "Integrated preview"** menu can be used to set whether the Integrated preview is to be used as to date for content only (Page and Site Store) or whether it is also to be used for media (Media Store):

- **use for content:** With this setting, the Integrated preview is used as to date for content in the Page and Site Store:

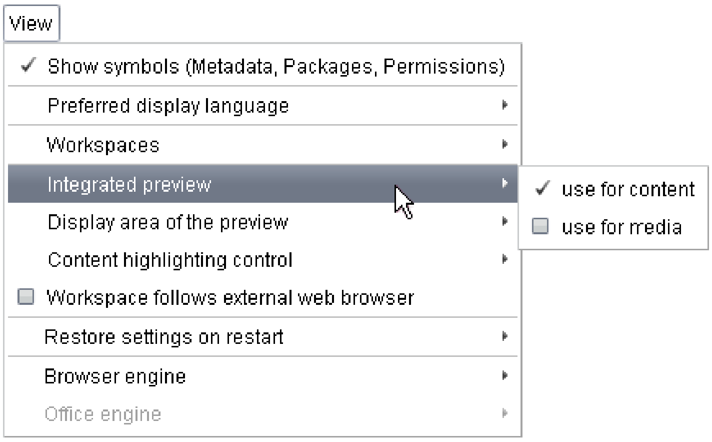

View
✓ Show symbols (Metadata, Packages, Permissions)
Preferred display language
Workspaces
Integrated preview
Display area of the preview
Content highlighting control
□ Workspace follows external web browser
Restore settings on restart
Browser engine
Office engine
✓ use for content
□ use for media

Figure 4-11: "View" / "Integrated preview" menu

In this case the Integrated preview is used to preview content, the default application entered in the operating system or the application which is selected under "User settings", "Editor" tab (under "Global settings") is used to preview and edit media.

- **use for content** and **use for media:** If Media Store media are also to be displayed and edited, as the case may be, in the Integrated preview, both options must be activated. This can be used to select whether Microsoft Office, OpenOffice (currently BETA test stage only!) or Google Docs (currently BETA test stage only!) is to be used to display and edit or process media from the Media Store. If the "disabled" entry is used, Integrated preview is not used to display media. See also below.
- **use for media:** If the Integrated preview is to be used for Media Store media only, the "use for media" option can be enabled. In this case, the default browser selected in the "User settings", "Browser" tab (under "Global settings") or entered in the operating system is used for the preview of content. The menu entry "Content highlighting control" is then greyed out.
- If **both options are disabled**, the default browser selected in the "User settings", "Browser" tab (under "Global settings") or entered in the operating system is used for the preview of content, the default application entered in the operating system or the application which is selected under "User settings", "Editor" tab (under "Global settings") is used to preview media.

If the "use for media" option is enabled, the **"View" / "Office engine"** menu can be used to set which application is to be used for Microsoft Office and OpenOffice file formats. The respective application must be installed on the workstation.

- **Microsoft Office (Windows only):** If this option is enabled, the relevant Microsoft Office application is used to display und edit office documents in the Integrated preview. This menu item is not displayed if JavaClient is operated in an operating system other than Windows.

- **OpenOffice (BETA, not MacOS):** If this option is enabled, the relevant OpenOffice application is used to display and edit office documents in the Integrated preview. This menu item is not displayed if JavaClient is operated for Mac OS.
- **Google Docs (BETA):** If this option is enabled, Google Docs is used to display and edit office documents in the Integrated preview.

> ! *The OpenOffice and Google Docs integration are currently in the BETA test stage and are not released officially yet!*

- **disabled:** If this option is enabled, neither Microsoft Office nor OpenOffice is used to display office documents; instead, they continue to be opened in an external application as to date.

If the option "use for media" is activated, you can decide via the menu **"View" / "Graphic engine"**, which application is to be used for which picture file format.

- **Java Image Editor:** If this option is activated, the Java Image Editor is used for displaying and editing pictures (see Chapter 4.2.3.2 page 42).
- **Simple image processing (Picnik):** If this option is activated, the online image processing service www.picnik.de is used for displaying and editing pictures (see Chapter 4.2.3 page 40).
- **Enhanced image processing (Pixlr):** If this option is activated, the online image processing service www.pixlr.com is used for displaying and editing pictures (see Chapter 4.2.3 page 40).
- **disabled:** If this option is activated, the image processing functions, well-known in FirstSpirit, are available (see Chapter 4.2.3.1 page 42).

> ! *Depending on the project administrator's settings and the operating system on which the JavaClient is currently operated, all menu items explained here can be greyed out or not available.*

4.2.2.4 Display of the FirstSpirit Online Help

While to date, FirstSpirit Online Documentation (ODFS) was only displayed in an external browser, it can now be opened in the Integrated preview. The search option in ODFS can also be used in the Integrated preview. The relevant documentation is displayed when Help is opened, depending on the context in which the user currently is. When working in the Integrated preview, when Help is opened, e.g. the "Page Store of the JavaClient" chapter in the *FirstSpirit Manual for Editors (JavaClient)* is opened in the internal preview.

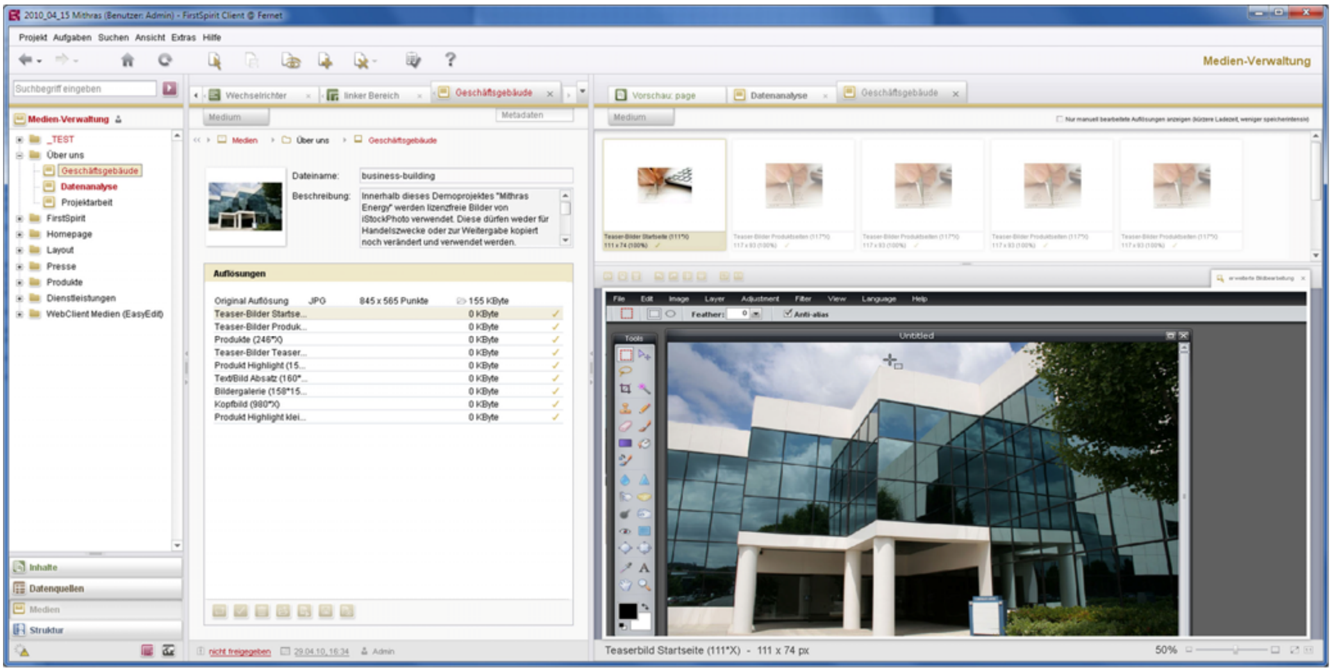
FirstSpirit Online Help can be displayed using F1, the ? icon of the JavaClient toolbar or the "Help" menu of the JavaClient menu bar. To do this, the Integrated preview must be enabled for content ("View" menu / "Integrated preview" / "use for content").

4.2.3 Image editing in the AppCenter

The cropping functions for Media Store images within JavaClient (via the "Edit image cutout" function in the workspace of an image) are added to in Version 4.2R4. These new functions are provided in the AppCenter via special image editing applications ("editors"), offer enhanced editing functions and – depending on the project administrator's settings (see Chapter 6.2.4 page 130) can be activated using the

- Java Image Editor
- Simple image editing (Picnik) and
- Enhanced image editing (Pixlr)

menu entries in the "View" / "Graphic engine" menu. These editors differ with regard to how they are operated and their functional scope. Picnik and Pixlr are online image editing services, which access www.picnik.com and www.pixlr.com respectively. Picnik enables simple, intuitive modification of photos; Pixlr is more orientated to professional image editing software such as Adobe Photoshop. An active internet connection is required to use these editors. Further information is given on the relevant internet sites. The Java Image Editor, on the other hand, is an integrated Java application, which does without internet access and special software installed on the workstation. Its range of functions is also based on that of professional image editing software.



Dateiname: business-building

Beschreibung: Innerhalb dieses Demoprojektes "Mithras Energy" werden lizenzfreie Bilder von iStockPhoto verwendet. Diese dürfen weder für Handelszwecke oder zur Weitergabe kopiert noch verändert und verwendet werden.

Auflösungen

Original Auflösung	JPG	845 x 565 Punkte	155 KByte	
Teaser-Bilder Startse...			0 KByte	✓
Teaser-Bilder Produk...			0 KByte	✓
Produkte (246*X)			0 KByte	✓
Teaser-Bilder Teaser...			0 KByte	✓
Produkt Highlight (15...			0 KByte	✓
Text/Bild Absatz (160*...			0 KByte	✓
Bildergalerie (158*15...			0 KByte	✓
Kopfbild (980*X)			0 KByte	✓
Produkt Highlight klei...			0 KByte	✓

Figure 4-12: JavaClient – enhanced image editing

For each Media Store image, a bar ("ribbon") with the different resolutions of the image is displayed at the top in the AppCenter (for this, the "Integrated preview for media" option must be activated in the "View" / "Integrated preview" menu, see also Chapter 4.2.2.3 page37). If more resolutions exist than can be displayed in the current ribbon height, it is possible to vertically scroll down the ribbon. If necessary, the ribbon can be collapsed. The original resolution, as well as manually edited resolutions, is displayed with a white background, the unedited resolutions automatically calculated by the system are displayed faded. Here too, as with the familiar resolution list used to date in the workspace (middle area of JavaClient), image variants whose resolution does not match the specified resolution are marked with the ❗ icon. The editing functions of the selected editor always relate to the resolution selected in the ribbon (yellow border). Below this is the area in which the image can be edited using the respective available functions.

For each image to be displayed or edited in the AppCenter, a separate tab must be opened, by changing to edit mode ("Edit on/off" context menu, Icon "Switch to View mode" of the toolbar or Ctrl + E) or request a preview ("Preview" context menu, "Preview" icon of the toolbar or Ctrl + P).

If "Edit" mode is activated for an image, the familiar editing functions used to date are initially available (▢ icon in the workspace) (see Chapter 4.2.3.1 page 42).

The "Enhanced image editing" button is used to start enhanced image editing, using the Editor, set using the "View" / "Graphic engine" menu. The functions, which were available via the icon, are then no longer available. If the "Enhanced image editing" button is clicked again, the program returns to simple image editing mode.

4.2.3.1 Simple image editing

The icon in the AppCenter must be clicked to start simple image editing using the functions available to date in the workspace. Only the currently selected image / the selected resolution can be edited. If another resolution is to be edited, the icon must be clicked again; only then can another resolution be selected.

The tab "Preview" in the working space can be displayed or hidden if needed by updating via F5 or the respective icon in the symbol bar.

4.2.3.2 The Java Image Editor

To edit an image / resolution using the Java Image Editor, the "Java Image Editor" menu item must be selected in the "View" / "Graphic engine" menu. The image to be edited in the Media Store must be set in Edit mode. The required resolution in the ribbon must be selected by clicking it with the mouse, so that the "Enhanced image editing" button becomes active.

After clicking this button, the editor with its editing functions and tools and the image file with the resolution selected in the ribbon are displayed in the editing window below the ribbon:

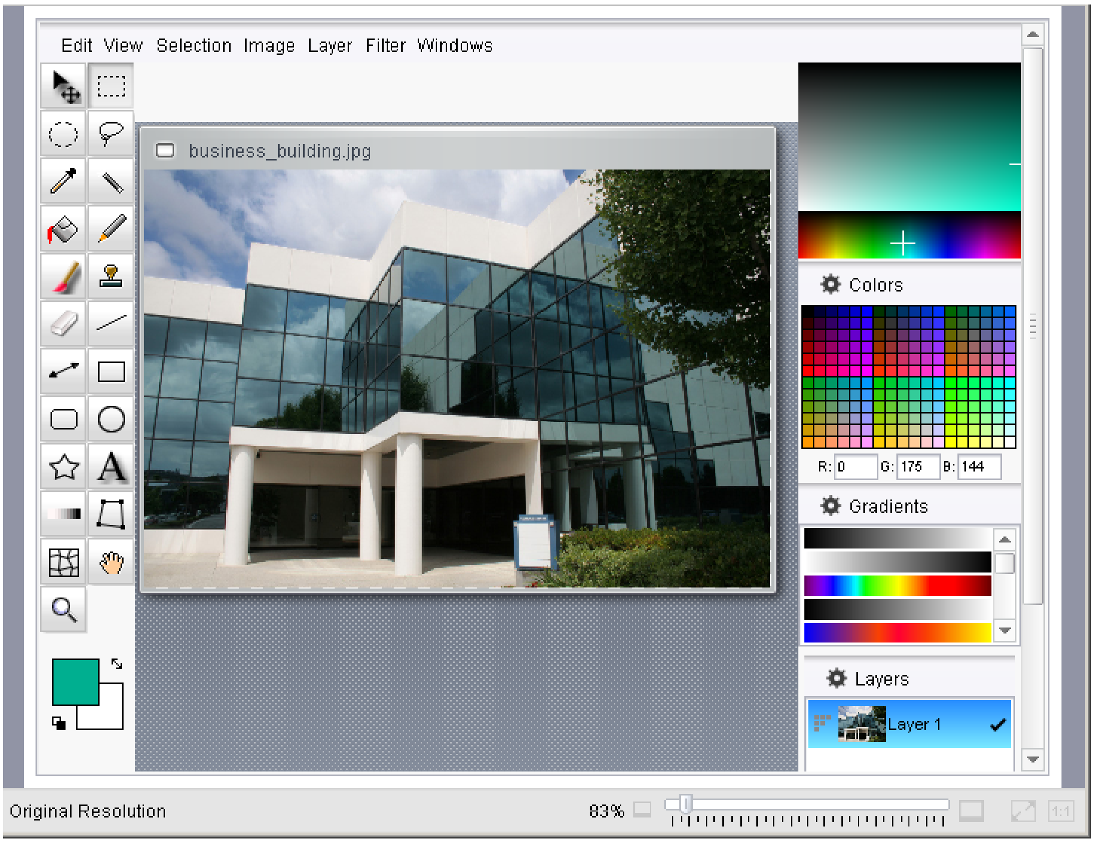


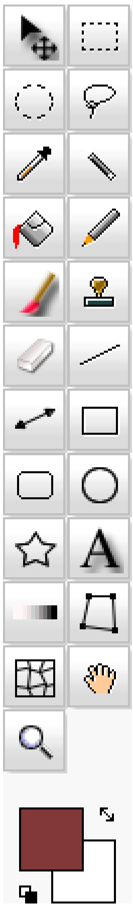
Figure 4-13: Enhanced image editing – Java Image Editor

The image can be edited using the available functions. These are comparable with those of other relevant image editing programs.

> *Several menus, tooltips and configuration dialogs (e.g. for filters) cannot be localised for technical reasons and are therefore only available in English.*

> *Several of the keyboard shortcuts displayed in the menus do not work for the editor, but instead perform the function known to date in JavaClient.*

4.2.3.2.1 Toolbar



Some tools provide enhanced configuration options. These are displayed above the image, below the menu bar.

Selection tools

This icon can be used to move a selection or to resize it using the selection grab handles.

Use these icons to select a square / rectangular, round / oval or lasso for user-defined shape of the image for further editing. Such a selection can, e.g. be applied using the "Edit" menu, modified using the "Selection" menu and moved using the icon. All painting tools (see below) are only applied within a selection. Additional functions are available with the Ctrl key pressed.

This icon can be used to select contiguous areas, which have the same or similar colour.

Use this icon to select through the colour of pixels in the image.

Painting and drawing tools

Use this icon to fill adjacent areas with similar colour.

This icon can be used to draw a hard-edged freehand line.

This icon is used to draw softer coloured lines.

Use this icon to duplicate a defined area of the image.

Use this icon to make pixels of the image transparent.

These icons can be used to create straight lines and arrows with defined width and fill method.

These icons can be used to create filled or unfilled squares / rectangles, with angular or rounded corners, circles / ovals or stars.

View

If the image window has scroll bars due to the zoom level set, this icon can be used to move the image within the window.

Use this icon to increase / reduce (Ctrl) the zoom level (zoom in/out).

Other

Use this icon to insert a layer with text into the image.

This icon is used to add colour gradients.

These icons are used to modify the perspective of the image by moving the corner points or to distort the image using a grid.

These icons are used to display the current foreground and background colour. They can be swapped using the arrow icon. Use the black&white icon to select black and white as the background/foreground colour.

4.2.3.2.2 "Edit" menu

This menu is used to apply selections: They can be

- cut (**Cut**)
- copied (**Copy**)
- inserted (**Paste**)
- deleted (**Clear**)

- If several layers exist on top of each other, **Copy Merged** is used in the selected area to create a copy of all visible layers reduced to one layer.
- In addition, editing steps can also be undone (**Undo**).

4.2.3.2.3 "View" menu

This menu is used to control the view of the image:

- Use the **Zoom In** and **Zoom Out** entries to gradually zoom the image.
- Use the **1:1**, **2:1**, **4:1**, **8:1** and **16:1** entries to set a specific zoom level.
- The **Show Grid** entry display a grid overlay on the image.
- Use the **Show Selection as Mask** entry to display the selection as a mask.

4.2.3.2.4 "Selection" menu

Use this menu to modify an existing selection:

- **Select All:** Selects the whole image.
- **Select None:** Empties the selection so that there is no longer any selection in the image.
- **Invert:** Selects the area of the image previously not selected.
- **Feather…:** Creates a selection with soft edges. The radius can be specified.

- **Grow:** Increases the selection by the specified radius.
- **From Layer Transparency:** Selects all transparent areas of the image.
- **New Layer from Selection:** Creates a layer on the basis of the selection.

4.2.3.2.5 "Image" menu

Use this menu to modify the whole image:

- **Crop:** Removes all parts of the image except for the selection.
- **Image Size…:** This can be used to change the size of the image.
- **Flip Horizontal / Vertical / Diagonal:** Mirrors the image horizontally / vertically along the vertical axis or diagonally about the transverse axis.
- **Rotate 90 / -90 / 180 / …:** Rotates the image through 90° in the clockwise or anticlockwise direction, or through 180° in a clockwise direction or by the number of degrees defined by the user.
- **Fill Selection:** Fills the whole current selection with the selected foreground colour.

4.2.3.2.6 "Layer" menu

This menu is used to create, modify and delete layers. On exiting the Java Image Editor, several layers are merged to form one; the next time the Editor is started, only this level is available.

- **New Layer / via Copy / via Cut:** Creates a new empty layer, a layer with the content of the current selection or a layer with the current selection and removes the selection from the existing layer.
- **Delete / Duplicate Layer:** Deletes or duplicates the currently selected layer.
- **Rename…:** This menu entry can be used to change the name of the layer. The new name is not saved on exiting the Editor and is no longer available the next time the Java Image Editor is started.
- **Move Up / Down:** Moves the current layer one position up / down.
- **Merge Down:** Merges the currently selected layer with the layer(s) underneath it.
- **Flip Horizontal / Vertical / Diagonal:** Mirrors the selected layer horizontally / vertically along the vertical axis or diagonally about the transverse axis.
- **Rotate 90 / -90 / 180:** Rotates the selected layer through 90° in a clockwise or anticlockwise direction, or through 180° in a clockwise direction.

4.2.3.2.7 "Filter" menu

This menu can be used to apply filters to the image, the current layer or the current selection.

- **Repeat Last Filter:** Applies the last applied filter again.
- **Show Last Filter:** Displays the last applied filter or the corresponding configuration dialog.

4.2.3.2.8 "Windows" menu

This menu can be used to hide or show the palettes on the right-hand side.

4.2.3.2.9 Palettes

- **Colours:** The foreground colour can be selected here with a click.
- **RGB colours:** The foreground colour can be selected here by setting RGB values.
- **Swatches ("Colors"):** Here the foreground colour can be selected from a library. As a default, a palette of colours is used, which can be used, platform-independent, by all browsers ("web colours"). If necessary, you can import your own libraries (and save them again later). Uploaded colour libraries are not saved on exiting the Editor and are no longer available the next time the Java Image Editor is started.
- **Gradients:** A colour gradient can be selected here. The existing ones can be used, edited or the user's own gradients can be loaded. Uploaded or edited colour gradients are not saved on exiting the Editor and are no longer available the next time the Java Image Editor is started.
- **Layers:** New layers can be created, edited or deleted here. See also Chapter 4.2.3.2.6 page 47. In addition, fill methods can be set for the current layer. On exiting the Java Image Editor, several layers are merged to form one; the next time the Editor is started, only this level is available.

4.2.3.3 Saving edited images

The editors provide their own save functions. These must be used initially to save edited images / resolutions. However, in order for the changes to be finally accepted in the Media Store, the selected image must also be saved again in the Media Store.

- Via started editor:

  Changes to images / resolutions have to be saved first using the icon. The image / resolution can then be further edited. Finally, the changes must be accepted using the toolbar icons "Save" or "Switch to View mode", "Lock/Unlock" in the context menu or Ctrl + S or Ctrl + E.

- Java Image Editor:

  Changes to images / resolutions have to be saved first using the icon. The image / resolution can then be further edited. Finally, the changes must be accepted using the toolbar icons "Save" or "Switch to View mode", "Edit on/off" in the context menu or Ctrl + S or Ctrl + E.

- Picnik:
  Changes to images / resolutions have to be saved first using the "Save in FirstSpirit" button. The image / resolution can then be further edited. Finally, the changes must be accepted using the toolbar icons "Save" or "Switch to View mode", "Edit on/off" in the context menu or Ctrl + S or Ctrl + E.

- Pixlr:
  Changes to images / resolutions have to be saved first using the "Save..." menu entry in the "File" menu. The image / resolution can then be further edited. Finally, the changes must be accepted using the toolbar icons "Save" or "Switch to View mode", "Edit on/off" in the context menu or Ctrl + S or Ctrl + E.

4.2.4 Content Store: Handling of filtered data records

From Version 4.2R2 on, data sources can be filtered by means of different options and the display of data records be restricted. Some optimisations have been realised in this context with Version 4.2R4.

> ! *While **up to 4.2R2** the permission "Create object" was required for creating filters (context menu "Extras" / "Set filter" on data source level), **from 4.2R4** the permission "Create folder" is required.*

- **Filter settings operate restricting:** From 4.2R4 the search results of the following filter options are AND-related:
  - persistent filter via context menu "Extras" / "Set filter" on the data source

  - o temporary filter via the link "Set filter" above the table
  - o temporary filter via the link "Search" above or the icon "Data search" beneath a table.

  This means that when selecting more than one filter, the search result will be continuously restricted. Previously, an existing filter was overridden when selecting another one.

- **Release displayed lines:** Up to and including Version 4.2R2, the "Release all rows" function was available under "Extras" in the context menu on data sources (in projects working with the release option). With this, all data records of the data source concerned were simultaneously released, current filter settings were not taken into account. With Version 4.2R4, this function has been replaced by the "Release displayed lines" function. With this, only the data records of the data source displayed, depending on the current filter settings of the data source, are now released. All available filter settings are taken into account:
  - o persistent filter via context menu "Extras" / "Set filter" on the data source
  - o temporary filter via the link "Set filter" above the table
  - o temporary filter via the link "Search" above or the icon "Data search" beneath a table,
  - o temporary quantity restriction via the combobox above the table.

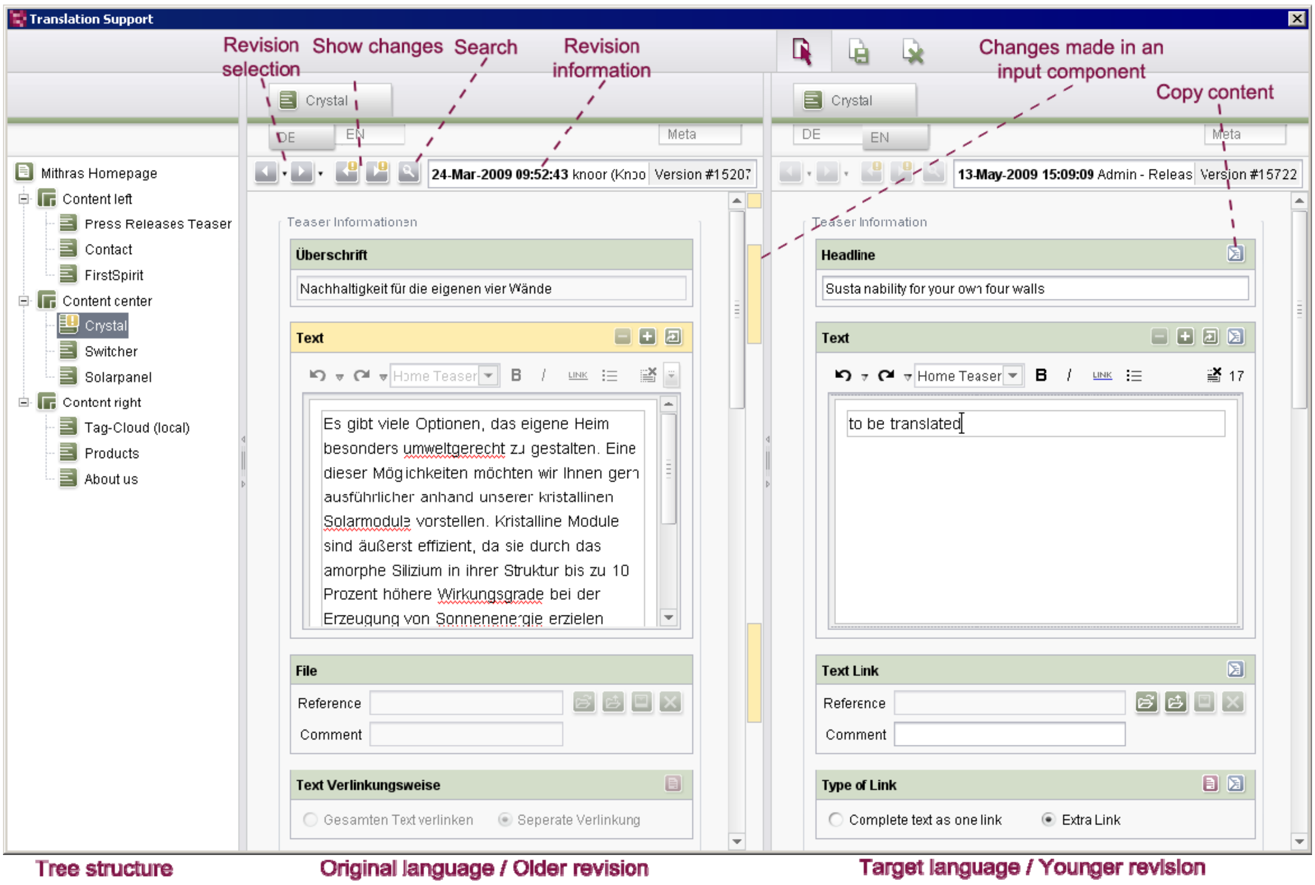
- **Jump to filtered data records:** If for example from an input component should be jumped to a data record which is not displayed in the overview because of the current filter settings the following notice will be shown: "The desired data record could not be found in the current view. Would you like to execute a search for it?"

4.2.5 Revision of version comparison and translation help

The dialog for version comparisons (context menu entry "Version history" / "Compare" button on any object) and the translation help ("Extras" menu, "Translation help" entry on pages and sections of the Page Store, data records of the Content Store and on Global contents) has been visually revised in Version 4.2R4.

The translation help is opened using the "Extras" menu, "Translation help" menu item. In Version 4.2R4, this menu item is now only active on the nodes of the tree structure on which the translation help can also be executed. The version history can be opened on each node of the tree structure, e.g. via Ctrl + H, the status bar or the context menu. The "Edit mode" combobox in the dialog for switching between comparison and translation mode has been removed in Version 4.2R4, as has "Post-translate" mode.

Opened on a section, the dialog could now look like this:



24-Mar-2009 09:52:43 knoor (Knoo Version #15207

Nachhaltigkeit für die eigenen vier Wände

Es gibt viele Optionen, das eigene Heim besonders umweltgerecht zu gestalten. Eine dieser Möglichkeiten möchten wir Ihnen gern ausführlicher anhand unserer kristallinen Solarmodule vorstellen. Kristalline Module sind äußerst effizient, da sie durch das amorphe Silizium in ihrer Struktur bis zu 10 Prozent höhere Wirkungsgrade bei der Erzeugung von Sonnenenergie erzielen

○ Gesamten Text verlinken ● Seperate Verlinkung

13-May-2009 15:09:09 Admin - Releas Version #15722

Susta nability for your own four walls

to be translated

○ Complete text as one link ● Extra Link

Figure 4-14: Version history / translation help – Page Store

The tree structure is now shown in a separate window on the left-hand side, as familiar from other dialogs. The input components of the selected object are displayed in two other areas of the window, each in the available languages. For translations, the original language is displayed on the left and the target language in the last revision of the object on the right; for comparisons, the revisions to be compared are displayed.

Between the window areas, a bar shows which input components contain changes. Here, yellow marking indicates that data has been changed in the input component, red marking indicates that data has been removed and green marking that values have been added. A tooltip shows the name of the relevant input component and it is possible to jump directly to the relevant input component with a click.

Icons are used to visualise in the tree structure whether and what type of differences exist between the compared versions. E.g. a yellow exclamation mark indicates that changes exist with respect to the content of the respective object (e.g. section or page), a green plus indicates that a node has been added, a red cross that a node has been deleted. Within the input components, changes between both versions are visualised by a coloured header row. Added text in DOM Editor and DOM table has a green background, text which has been removed is shown with a red background.

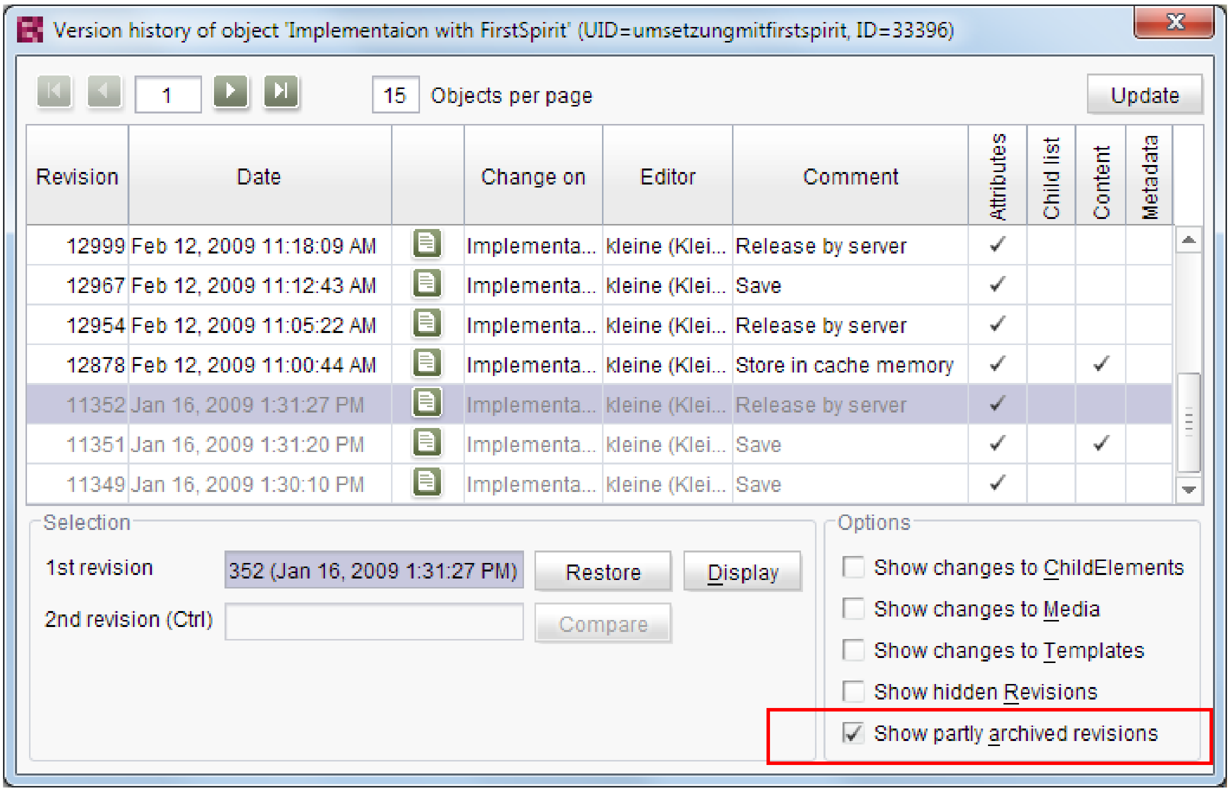
Above the input components the respective revision is displayed with the version number, date, time, last modifier and the comment assigned to the revision ("Revision information"). The Next and Back icons can be used to switch to the previous or next revision ("Revision selection") or the previous or next change ("Show changes") to the respective object.

Two different versions of an object in the same language are displayed in the window areas for **comparisons**. It is not possible to edit content in comparison mode.

For **translations**, existing content in the original language can be copied into the target language with the help of the 🗎 icon in the respective input component, in order to have a direct translation draft. The translation can then be made in this dialog, or as usual in the section concerned or the page concerned in the working space of the JavaClient. Each time "Save" is pressed (or Ctrl + S) or the "Switch to View mode" icon (Ctrl + E), a new revision is generated with the comment "Translated". Original language content cannot be edited.

4.2.6 Version history: New "Show partly archived revisions" option

In the version history dialog ("Version history" item in the context menu on a node), there is now a new "Show partly archived revisions" option:



Version history of object 'Implementaion with FirstSpirit' (UID=umsetzungmitfirstspirit, ID=33396)

Revision	Date		Change on	Editor	Comment	Attributes	Child list	Content	Metadata
12999	Feb 12, 2009 11:18:09 AM		Implementa...	kleine (Klei...	Release by server	✓			
12967	Feb 12, 2009 11:12:43 AM		Implementa...	kleine (Klei...	Save	✓			
12954	Feb 12, 2009 11:05:22 AM		Implementa...	kleine (Klei...	Release by server	✓			
12878	Feb 12, 2009 11:00:44 AM		Implementa...	kleine (Klei...	Store in cache memory	✓		✓	
11352	Jan 16, 2009 1:31:27 PM		Implementa...	kleine (Klei...	Release by server	✓			
11351	Jan 16, 2009 1:31:20 PM		Implementa...	kleine (Klei...	Save	✓		✓	
11349	Jan 16, 2009 1:30:10 PM		Implementa...	kleine (Klei...	Save	✓			

Selection

1st revision: 352 (Jan 16, 2009 1:31:27 PM)

2nd revision (Ctrl):

Options

- □ Show changes to ChildElements
- □ Show changes to Media
- □ Show changes to Templates
- □ Show hidden Revisions
- ☑ Show partly archived revisions

Figure 4-15: Version history with new option

Following archiving (see *FirstSpirit Manual for Administrators*, "Archiving old project statuses" chapter), as a default, so-called "partially archived revisions" are not displayed in the version history. To date, partially archived and hidden revisions could be displayed using the "Show hidden revisions" option. The new "Show partly archived revisions" option has been introduced to enable more intuitive use. With this option, all revisions of objects are shown which are still completely retained but whose revision number is smaller than the smallest/last not yet archived revision. This can be the case, e.g. if pages of the Page Store are archived, but not the page templates on which they are based (this is achieved using an appropriate schedule, in which the "content, media and data" options are enabled, the "templates" and "system data" options are disabled).

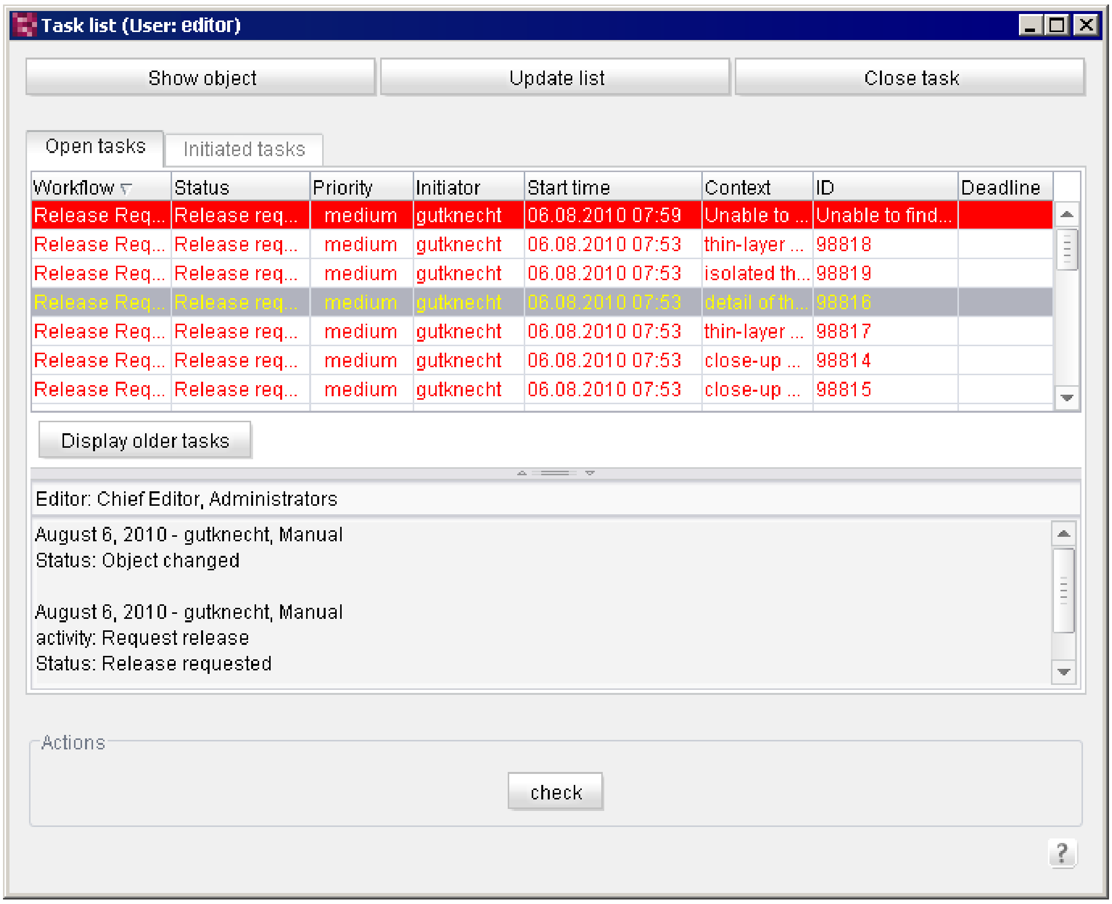
Partly archived revisions are shown in grey on activation of the option in the table. The buttons concerned can be used to perform the "Restore", "Display" and

"Compare" functions on them.

4.2.7 Changes to workflows and tasks

For performance reasons, from 4.2R4, only 25 tasks are initially displayed in the task list (Ctrl + T or "Tasks" menu / "Tasks list") on the "Open Tasks" and "Initiated Tasks" tabs. If more tasks exist, they can be shown using the **Display older tasks** button (see Figure 4-16). Initially, only 10 tasks are displayed on the personalised project entry page. Click the "Display older tasks" button to show a further 10 tasks. Click "Display all tasks" to open the Task list.

Invalid tasks, e.g. produced because an object on which a workflow is active is deleted, are visualised with a red background in the tasks list from 4.2R4 (see Figure 4-16). These cannot be set to the next status, they can only be closed using the "Close task" button. If the task can be repaired, e.g. the deleted object for which the workflow still exists is restored, the "Repair task" button appears. This is used to reset the task, the status colour and write protection.



Workflow	Status	Priority	Initiator	Start time	Context	ID	Deadline
Release Req...	Release req...	medium	gutknecht	06.08.2010 07:59	Unable to ...	Unable to find...	
Release Req...	Release req...	medium	gutknecht	06.08.2010 07:53	thin-layer ...	98818	
Release Req...	Release req...	medium	gutknecht	06.08.2010 07:53	isolated th...	98819	
Release Req...	Release req...	medium	gutknecht	06.08.2010 07:53	detail of th...	98816	
Release Req...	Release req...	medium	gutknecht	06.08.2010 07:53	thin-layer ...	98817	
Release Req...	Release req...	medium	gutknecht	06.08.2010 07:53	close-up ...	98814	
Release Req...	Release req...	medium	gutknecht	06.08.2010 07:53	close-up ...	98815	

Editor: Chief Editor, Administrators

August 6, 2010 - gutknecht, Manual
Status: Object changed

August 6, 2010 - gutknecht, Manual
activity: Request release
Status: Release requested

Figure 4-16: Task list with invalid tasks

4.3 New/Modified functions in WebClient

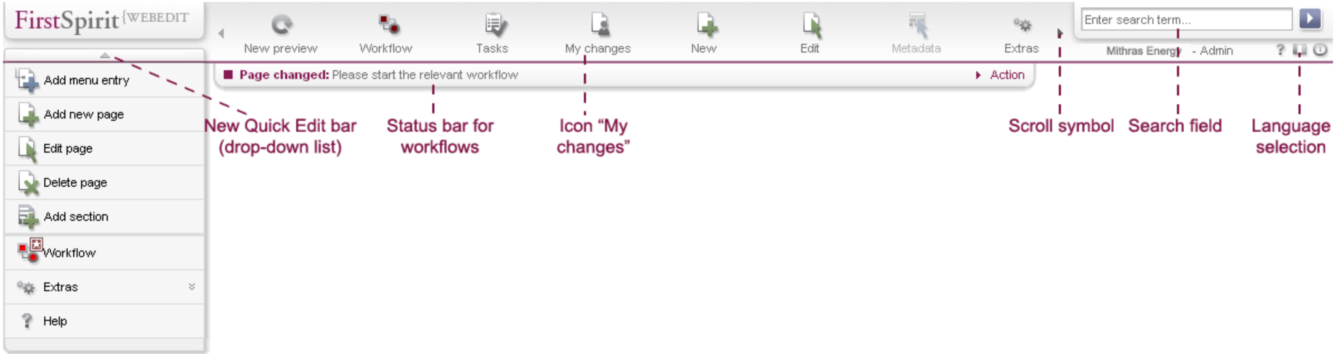
WebEdit has been visually revised again in Version 4.2R4, before being replaced by a completely new implementation in FirstSpirit Version 5.0. In addition, several new functions have been introduced. The user prompting is more or less unchanged.

> ! *From 4.2R4, only the "xp" theme will be supported. This is now the default theme. In addition, themes with customer-specific settings will no longer be supported.*

> ! *The new input component **FS_BUTTON** added to JavaClient (see Chapter 4.1.2 page 27) is currently not supported in the FirstSpirit WebClient. Appropriate support is planned for FirstSpirit Version 5. Only rudimentary support is available for the input component **FS_LIST** (see Chapter 4.1.1 page 25) in the WebClient in 4.2R4: at present, only the functions of the input component CMS_INPUT_CONTENTAREALIST (list for managing section contents) are available. More functions can be provided using the so-called applet mode. However, this will only be a transition solution; native FS_LIST support will exist from FirstSpirit Version 5.0.*

4.3.1 Revised WebEdit toolbar

In Version 4.2R4, the icons used in the WebEdit toolbar have been further adjusted to the familiar ones of FirstSpirit JavaClient. The lettering of the icons has been retained, but is now positioned below the icons:



Page changed: Please start the relevant workflow

Figure 4-17: Revised WebEdit toolbar

In addition, several new functions have been added to the Toolbar, namely

- **Scroll symbols:** If the WebEdit browser window is smaller than the WebEdit toolbar, the toolbar can be moved to the right or left (scrolled) using the ◀ and ▶

icons. The icons are only active if further toolbar functions can be displayed by scrolling to the left or right.

- **a drop-down list:** It contains the functions of the Quick-Edit bar at page level and should release it in a later version (see Chapter 4.3.2 page 56),
- **a search field:** This can be used to search the project by means of full text search (see also Chapter 4.3.3 page 57),
- **the "My changes" icon:** The user can use this function to have all pages of the project displayed which they have created or on which they have made changes. The **global search** function is used for this (see also Chapter 4.3.3 page 57),
- **a status bar for workflows:** Below the WebEdit toolbar, a status bar displays the current workflow status of the respective page (including changes to sections on the page and the corresponding page reference). Click the status bar to directly start or forward the workflow concerned (see Chapter 4.3.4 page 60).

In addition, the icons for language selection and Help have been move to the right-hand edge, below the search field.

4.3.2 New Quick-Edit bar at page level

From Version 4.2R4, the functions of the Quick-Edit bar have been replaced by the new vertical Quick-Edit bar. Click the double arrow icon to open a sub-menu below "Extras":

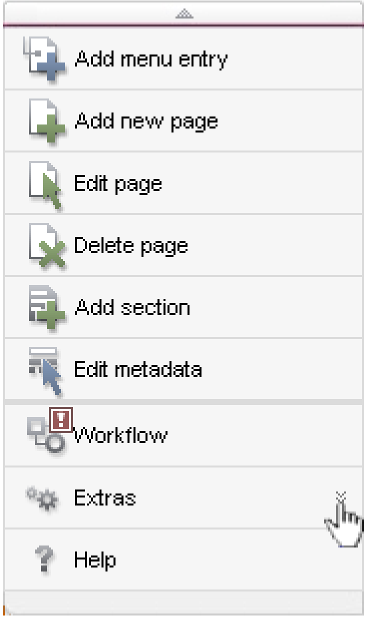


Figure 4-18: Vertical Quick-Edit bar

Functions are shown here according to the user's permissions. For reasons of compatibility, the familiar (horizontal) Quick-Edit bar continues to be supported. From Version 5.0, it will no longer be supported.

4.3.3 Global search

A search window is now available on each preview page of the WebClient for full text search within a project. It can be used to search through Site and Page Store. Search results can be output, filtered for a certain edit period and / or only the objects created or changed by the current user can be displayed (using the "My changes" icon of the WebEdit toolbar, see Chapter 4.3.1 page 55).

However, only the Page and Site Stores are searched. Hits from the Site Store are primarily output. Only if there is no page reference for a hit, the page concerned is output from the Page Store. Apart from texts on the project's pages, metadata and media referenced on the pages are also searched.

The search is started using the ▶ icon or <Enter>. The field can also remain empty. In this way, in conjunction with the available filters (e.g. chronologically or after current user, see below), e.g. a list of the last changes can be requested.

The hit list is displayed below the search window:

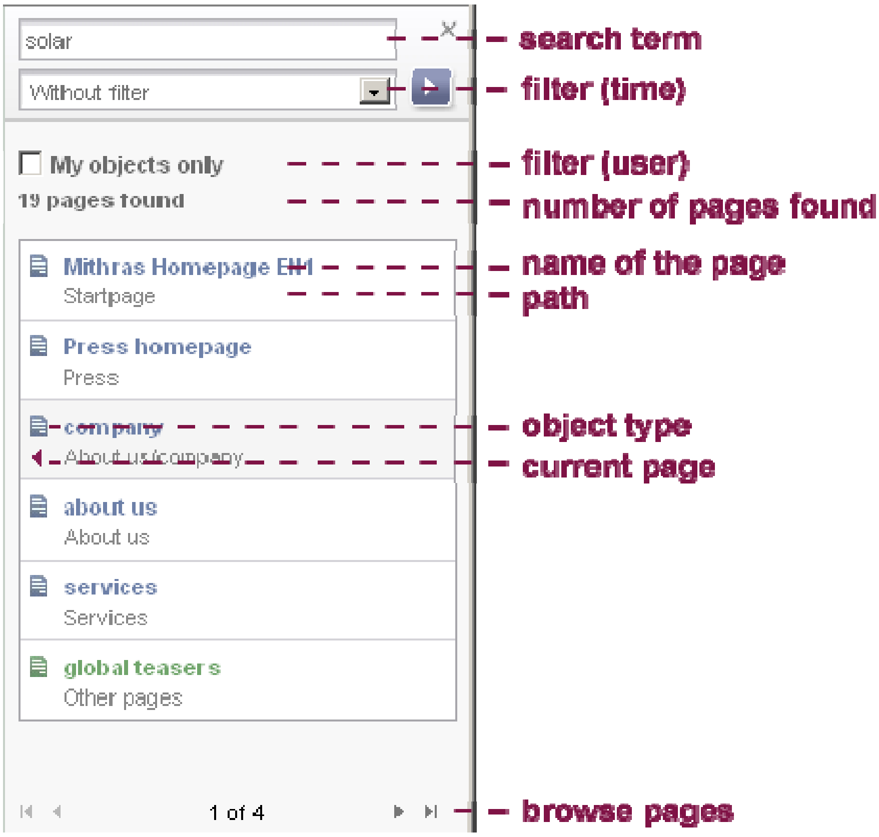
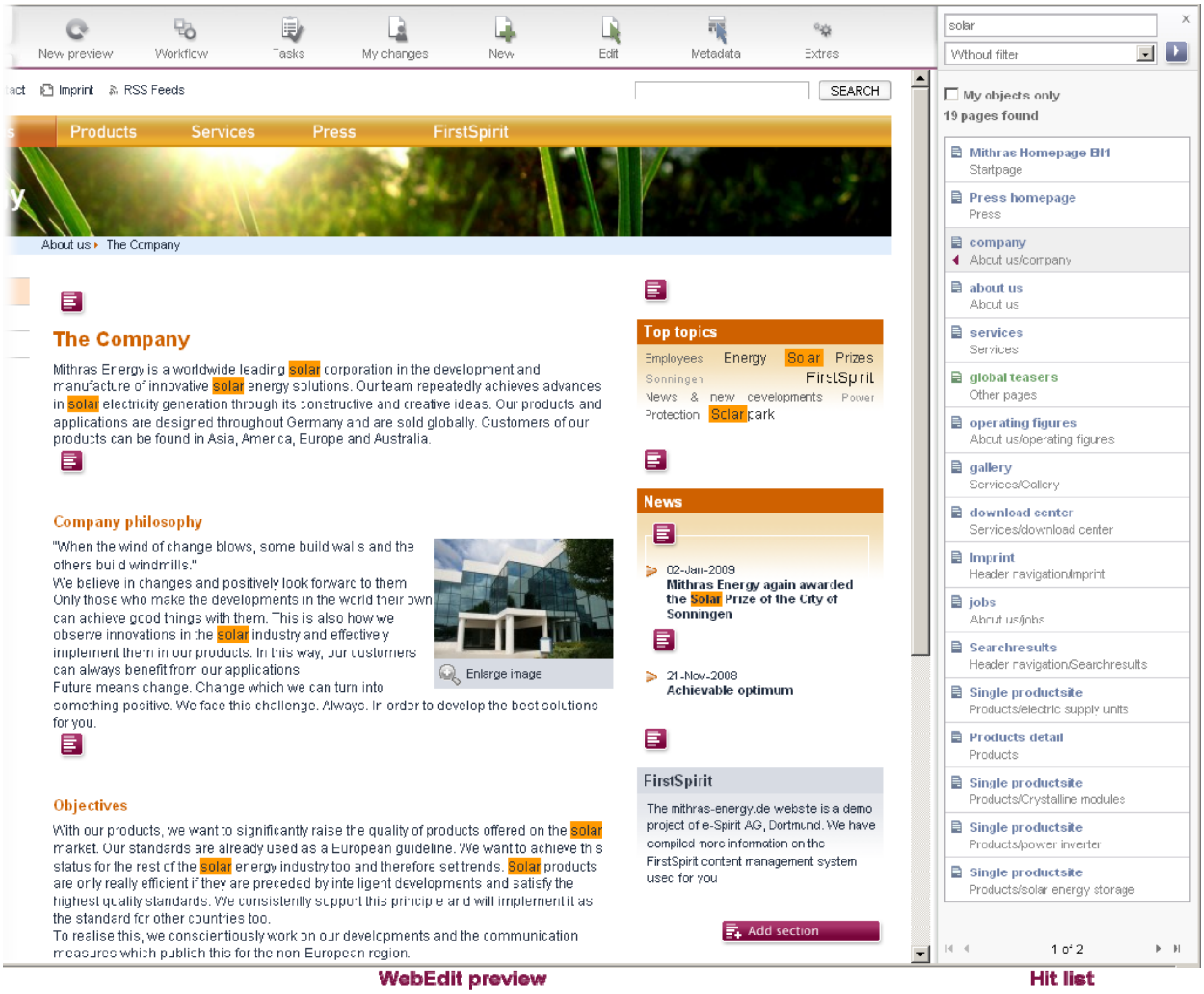


Figure 4-19: Hit list for the search term "solar"

The number of pages found is output via the hit list. If no results are found, the text "0 pages found" is displayed.

The hit list has a fixed width. If more objects are found than can be displayed in the available height, the icons below the hit list can be used to switch to the first/previous/next/last search results page.

The results are each displayed with their object icon (usually 🗎), name and path. Click a result to change the WebEdit preview to the page with the sought text.



About us ▸ The Company

The Company

Mithras Energy is a worldwide leading solar corporation in the development and manufacture of innovative solar energy solutions. Our team repeatedly achieves advances in solar electricity generation through its constructive and creative ideas. Our products and applications are designed throughout Germany and are sold globally. Customers of our products can be found in Asia, America, Europe and Australia.

Company philosophy

"When the wind of change blows, some build walls and the others build windmills."
We believe in changes and positively look forward to them. Only those who make the developments in the world their own can achieve good things with them. This is also how we observe innovations in the solar industry and effectively implement them in our products. In this way, our customers can always benefit from our applications.
Future means change. Change which we can turn into something positive. We face this challenge. Always. In order to develop the best solutions for you.

Objectives

With our products, we want to significantly raise the quality of products offered on the solar market. Our standards are already used as a European guideline. We want to achieve this status for the rest of the solar energy industry too and therefore set trends. Solar products are only really efficient if they are preceded by intelligent developments and satisfy the highest quality standards. We consistently support this principle and will implement it as the standard for other countries too.
To realise this, we conscientiously work on our developments and the communication measures which publish this for the non European region.

Top topics

Employees Energy Solar Prizes Sonningen FirstSpirit News & new developments Power Protection Solar park

News

02-Jan-2009
Mithras Energy again awarded the Solar Prize of the City of Sonningen

21-Nov-2008
Achievable optimum

FirstSpirit

The mithras-energy.de website is a demo project of e-Spirit AG, Dortmund. We have compiled more information on the FirstSpirit content management system used for you

19 pages found

- **Mithras Homepage EH** – Startpage
- **Press homepage** – Press
- **company** – About us/company
- **about us** – About us
- **services** – Services
- **global teasers** – Other pages
- **operating figures** – About us/operating figures
- **gallery** – Services/Gallery
- **download center** – Services/download center
- **Imprint** – Header navigation/Imprint
- **jobs** – About us/jobs
- **Searchresults** – Header navigation/Searchresults
- **Single productsite** – Products/electric supply units
- **Products detail** – Products
- **Single productsite** – Products/Crystalline modules
- **Single productsite** – Products/power inverter
- **Single productsite** – Products/solar energy storage

1 of 2

Figure 4-20: Search results page in the preview

The ◀ icon in the hit list indicates that the selected hit is located on the current page on the left in the WebEdit preview. The hit(s) is/are highlighted in colour in the preview.

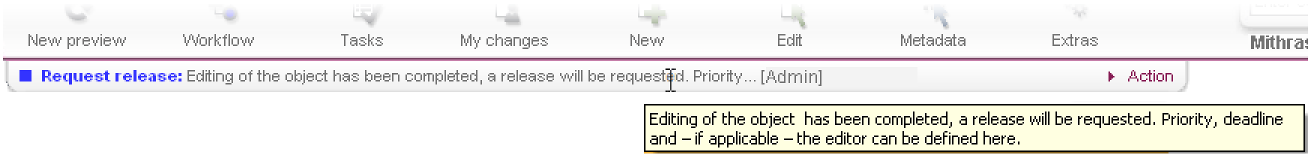
Filter: The search result can be limited in time with the help of filters. If the "Changed, last week" filter is selected from the drop-down list, only page references created or changed within the past week are displayed. If the "Changed, last month" filter is selected, only page references created or changed within the past month are displayed. These filters can be combined with the "My objects only" filter (see below). Click the Search icon ▶ to activate the filter. The default setting is "Without filter".

My objects only: This checkbox can be used to limit the search result to objects which have been created or changed by the currently logged in user. Click the Search icon ▶ to activate this filter. As a default, this checkbox is not selected (exception: Display of "My Changes", Chapter 4.3.1 page 55).

The hit list remains open until it is closed using the X icon on the right or a new preview is requested. If a search term has been entered, it remains until it is deleted, another search term is entered or a new preview is requested.

4.3.4 Status bar for workflows

A status bar for workflows has been introduced underneath the WebEdit toolbar in order to be able to identify better, which workflow status a page has and which step has to be performed next.



■ **Request release:** Editing of the object has been completed, a release will be requested. Priority... [Admin]

Editing of the object has been completed, a release will be requested. Priority, deadline and – if applicable – the editor can be defined here.

Figure 4-21: New Workflow status bar

The next step required to process the workflow is displayed here. This is displayed in the respective status colour defined by the project developer (here: "Request release"). Equally, the user who switched the page to its current status is displayed (here: "Admin") and an optional comment of the user, which they entered on forwarding the workflow. If the user did not enter any comment, a description stored for the status by the project developer is displayed instead. If the text is longer than the available space, the complete text can be given in a tooltip. This is shown when the mouse cursor is moved over the comment or description text.

The workflow can be forwarded by clicking "Action" on the right-hand side of the status bar. A dialog window opens, with which the next workflow step can be performed.

For detailed information on workflows in WebEdit, see FirstSpirit Manual for Editors (WebEdit), "Start Workflow / Workflow Action" chapter".

Unlike previous versions in which the icons concerned merely indicated that a workflow on the page had been started, the new status bar is now used to also visualise that it has been changed. Changes to the relevant page reference are also visualised.

The workflow status is also displayed in the same way as to date via the Workflow icons of the WebEdit toolbar and the Quick-Edit bar used to date and the new Quick-Edit bar at page level (see Chapter 4.3.2 page 56).

4.3.5 Revised selection dialogs

The following input components have a selection dialog in WebEdit, with which references to FirstSpirit objects can be created:

- File Selection (CMS_INPUT_FILE)
- Page Reference (CMS_INPUT_PAGEREF)
- Picture selection (CMS_INPUT_PICTURE)
- Data record selection (CMS_INPUT_OBJECTCHOOSER, FS_DATASET)
- Reference Selection (FS_REFERENCE)
- Links which enable the selection of FirstSpirit objects (CMS_INPUT_LINK)

These selection dialogs are opened using the "Open" icon (e.g. 📂). In Version 4.2R4, on the one hand the design has been revised, on the other the selection dialogs now also contain a search option – similar to the case in FirstSpirit JavaClient.

Exception: The search is not available for the Content Store.

This search can be used to search through the stores from which a reference can be selected, e.g. for images and files from the Media Store or page references from the Site Store. This search option makes it easier to work with larger projects and extensive content.

For the input component CMS_INPUT_FILE, the selection dialog search, e.g. looks like this:

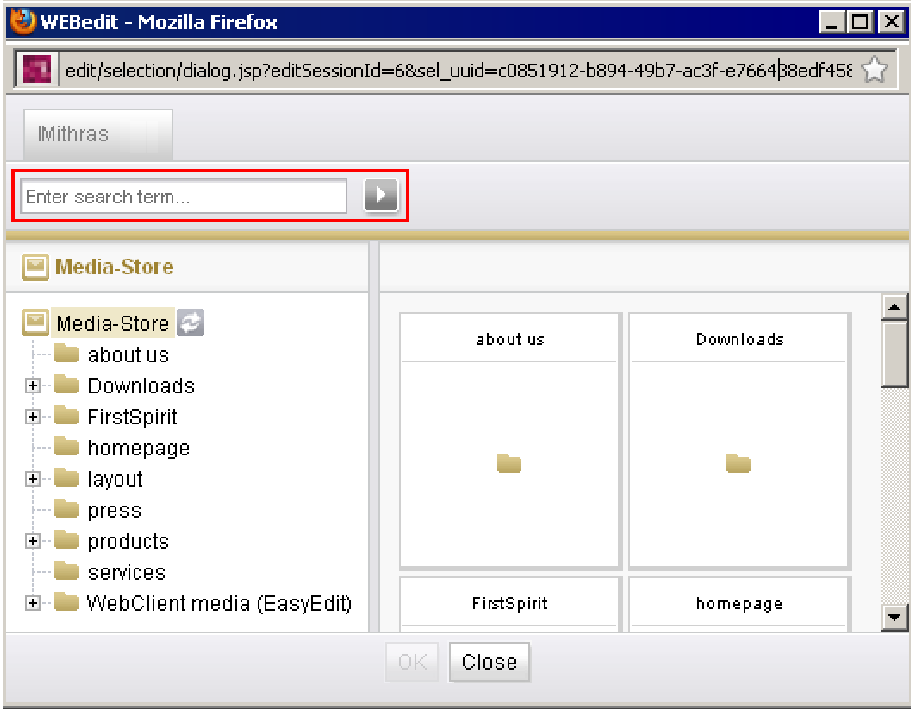
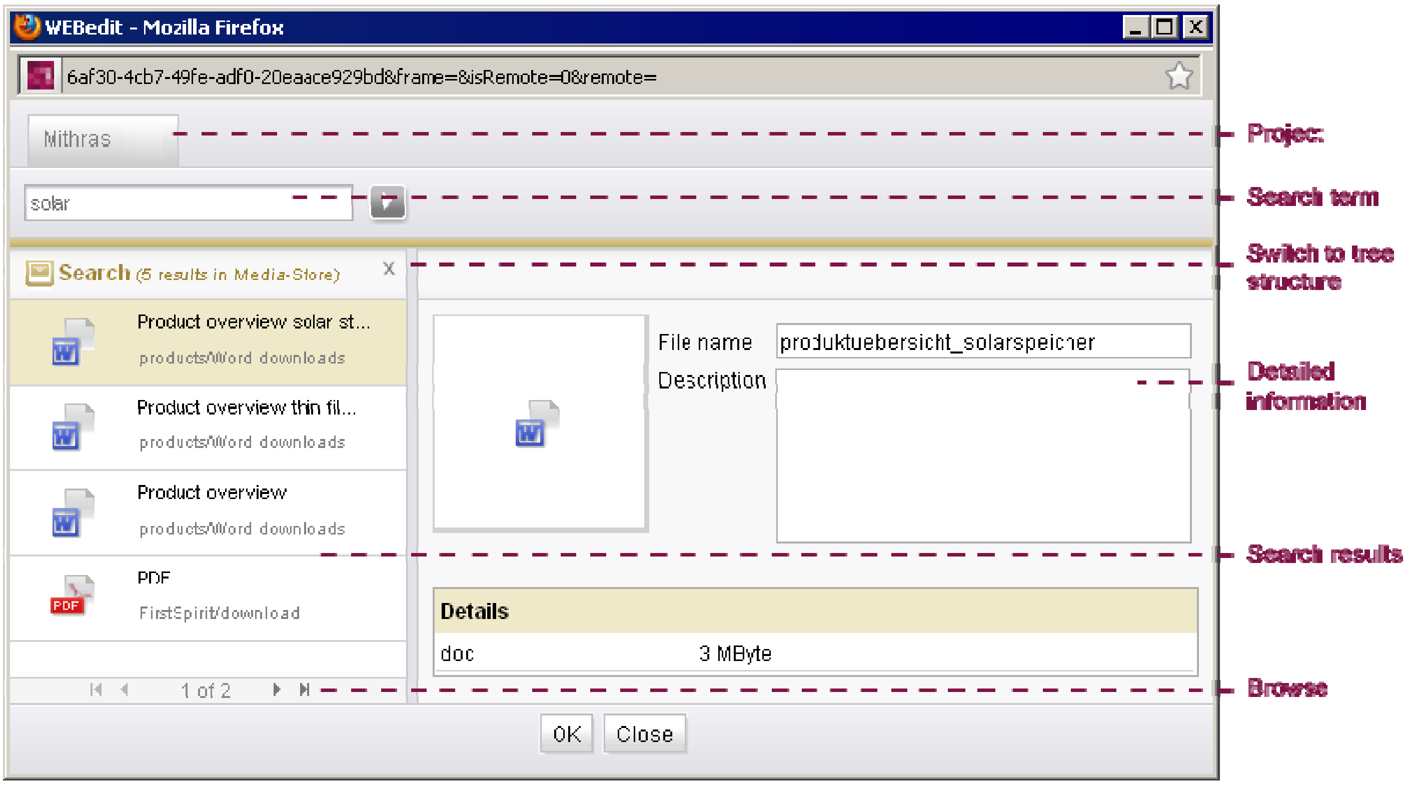


Figure 4-22: Search field in the File Selection dialog

Working with the search is exemplarily presented in the following for the input component CMS_INPUT_FILE. The search in the other input components essentially functions analogous to this.

The search field in the File Selection dialog can be used to search through the Media Store of the respective project for files by means of a full text search. Editorial content is also included, e.g. texts in PDF or Word files.

After entering the sought term, the search can be started using the arrow icon or <Enter>. In the following dialog the search results are displayed in the left-hand column. Click a result to have the detailed information shown in the left-hand column:



WEBedit - Mozilla Firefox

6af30-4cb7-49fe-adf0-20eaace929bd&frame=&isRemote=0&remote=

Mithras

solar

Search (5 results in Media-Store)

Product overview solar st...	products/Word downloads
Product overview thin fil...	products/Word downloads
Product overview	products/Word downloads
PDF	FirstSpirit/download

1 of 2

File name: produktuebersicht_solarspeicher

Description:

Details	
doc	3 MByte

Figure 4-23: Search results

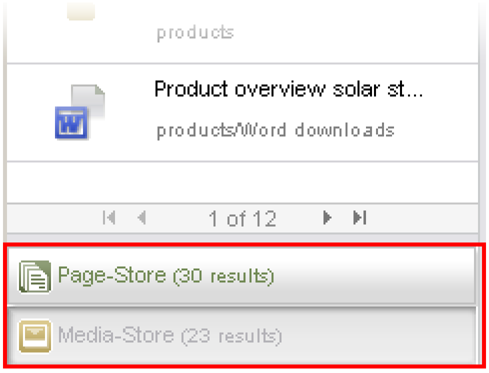
The results are listed with display name, file symbol (e.g. Word or PDF symbol) and path. The number of search results is displayed above the results list. The detailed information on the selected object is displayed on the right-hand side of the dialog.

If more results exist than can be displayed within the set height of the dialog, the arrow icons underneath the results list can be used to switch to other results pages.

Click the x icon next to the number of search results to display the tree structure of the Media Store instead of the results list. This is expanded up to the level on which the previously selected medium is located. This display can also be used for navigation through the Media Store without starting the search. If this icon has been pressed, it is no longer possible to switch to the search results list, if necessary, the search must be started again.

Press "OK" to copy the selected file into the input component.

If an input component enables selection from **several stores** (e.g. FS_REFERENCE or internal links), all stores are searched. The buttons at the bottom end of the dialog can be used to switch between the results of the two stores. The respective number of hits is shown there:


products

Product overview solar st...

products/Word downloads

1 of 12

Page-Store (30 results)

Media-Store (23 results)

Figure 4-24: Search results in the Site and Media Store

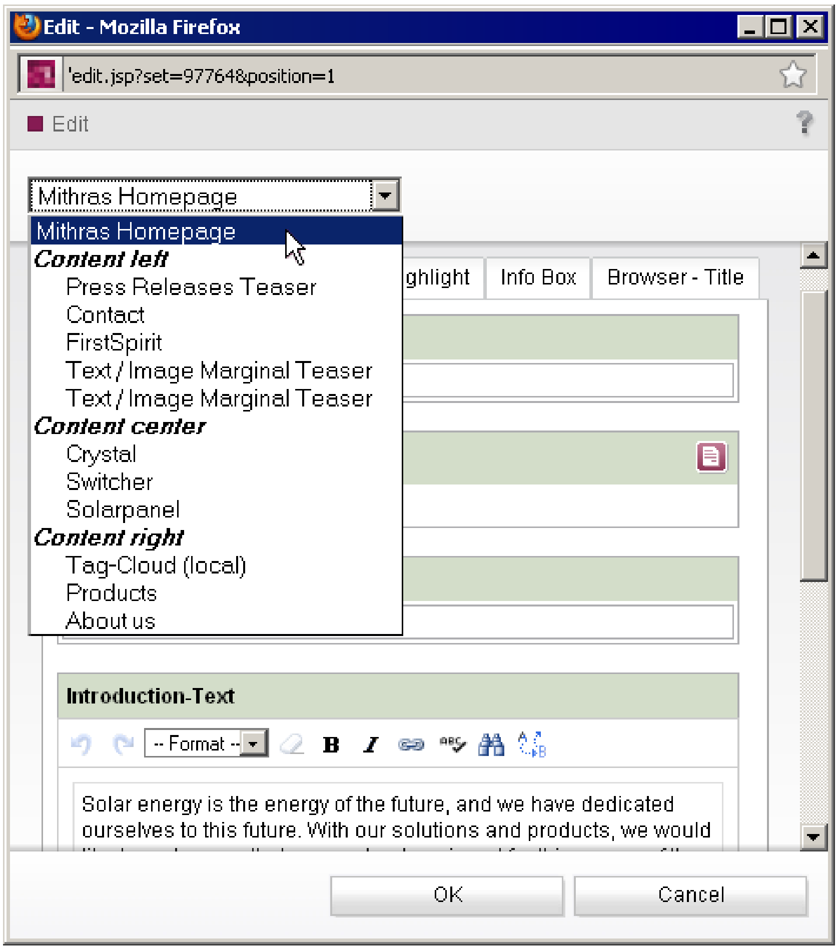
If access to one or several **remote projects** is configured for the input component, the remote projects are displayed as additional tabs next to the tab of the local project (in Figure 4-22: "Mithras"). If the Media Store is to be searched through for a remote project, switch to the respective tab and start the search from there.

For detailed information on working with this input component in WebEdit, see FirstSpirit Manual for Editors (WebEdit), "The standard entry elements" chapter, especially the sub-section on input component CMS_INPUT_FILE, "Select an existing file from the Media Store".

4.3.6 Revised "Edit" and "New" dialogs

Several dialogs, e.g. for creating or editing menu levels, pages or sections, have been slightly revised in Version 4.2R4. But the user prompting is more or less unchanged.

"Edit" dialogs, which previously required two-stage selection of the element to be edited, now contain only one drop-down list, from the desired element can be directly selected, e.g.



'edit.jsp?set=97764&position=1

Introduction-Text

Solar energy is the energy of the future, and we have dedicated ourselves to this future. With our solutions and products, we would

Figure 4-25: "Edit" dialog

Whereas before, you selected first whether the input components of the page or section were to be edited and then the desired section, the page or a specific section is now selected directly via the drop-down list. All input components available for the page or section selected via the drop-down list are displayed in the bottom part of the dialog window. This selection is used in an analogous way for data records too.

This concerns the dialogs which open via the

- "Edit" and "Metadata" icons in the WebEdit toolbar
- "Edit page" in the Quick-Edit bar

and the following dialog on selecting the

- "New" icon in the WebEdit toolbar.

Example "Metadata" dialog:

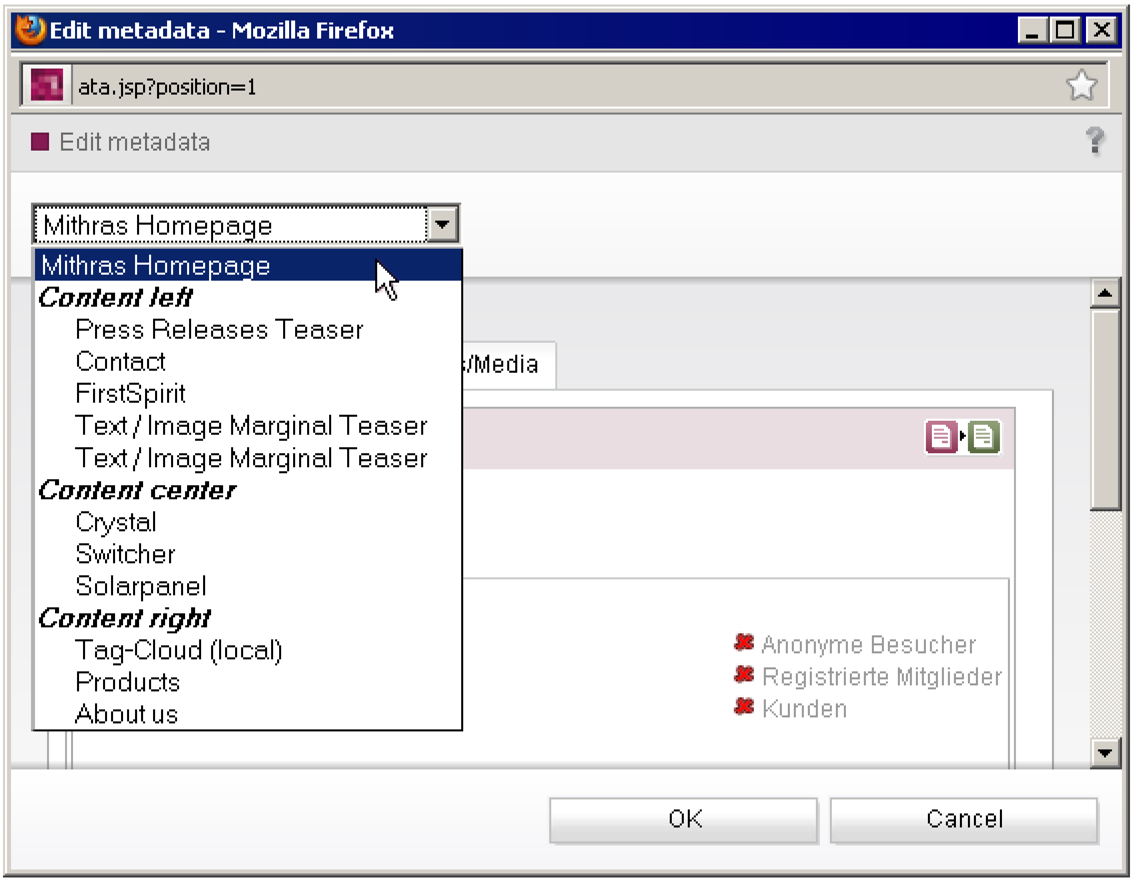
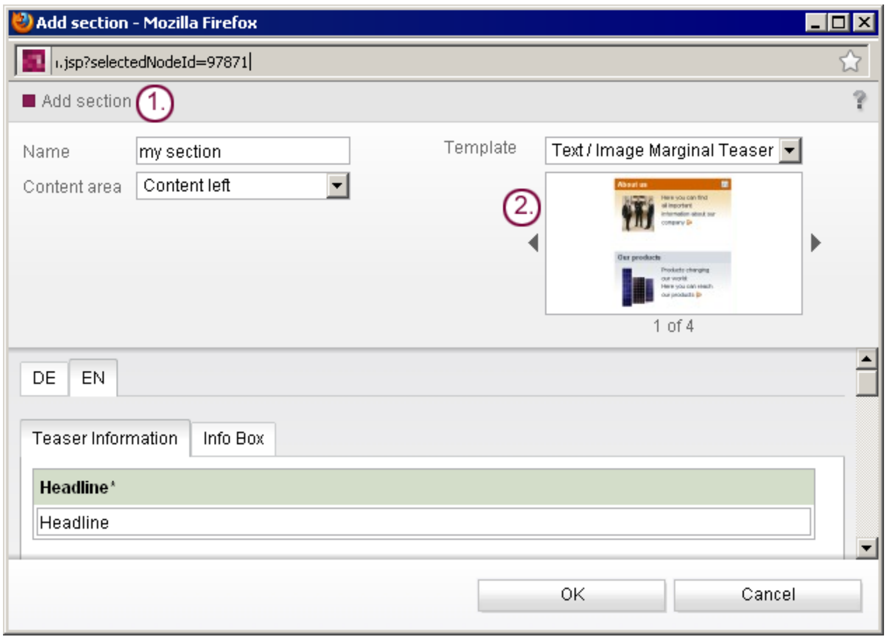


Figure 4-26: "Metadata" dialog

The "New" dialogs, which are opened via the Quick-Edit bar (horizontal or vertical), have been changed slightly with respect to their layout and now provide more convenient selection of the required template, e.g.



Name: my section

Content area: Content left

Template: Text / Image Marginal Teaser

Headline: Headline

Figure 4-27: "Add section" dialog

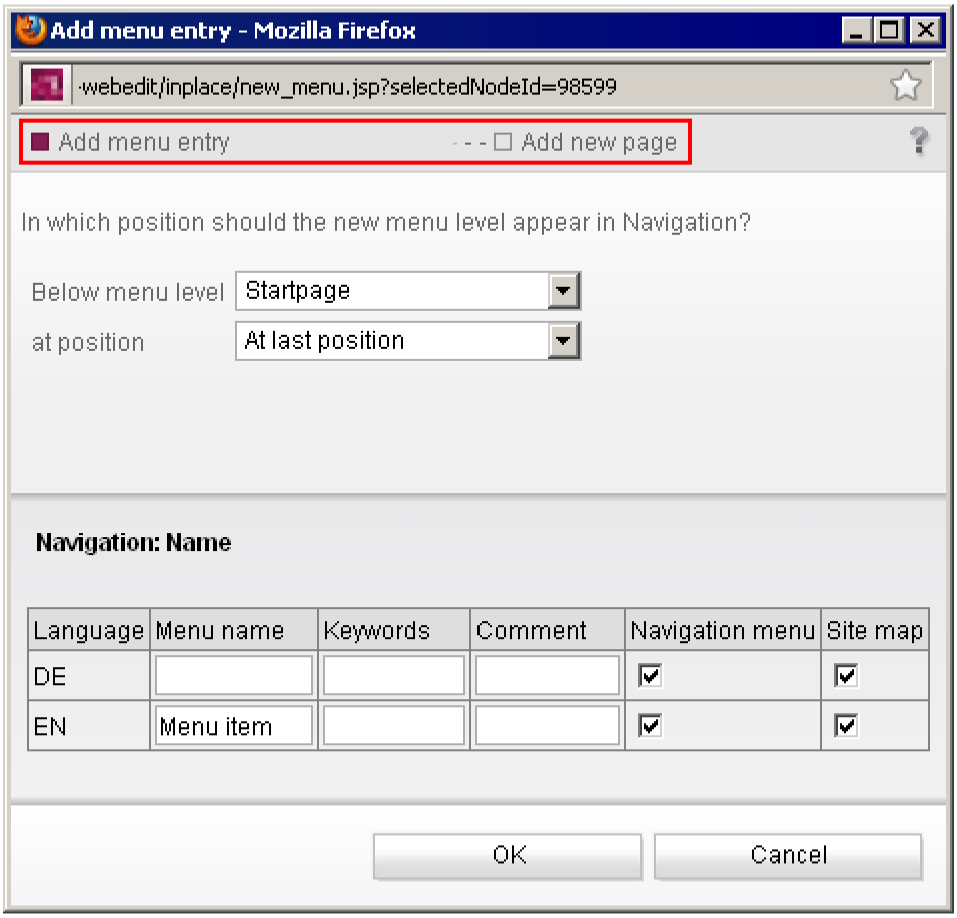
1. The action or dialog name is now always shown on the top left-hand side.

2. The scroll symbols can be used to choose between the available templates on the basis of the preview image (if provided by the template developer for the respective template). The corresponding input components are shown directly in the bottom area. Alternatively, the template can be selected by name as to date using the drop-down list.

This concerns the dialogs which are opened via

- the "Add menu entry", "Add new page" and "Add section" icons/buttons in the Quick-Edit bar (horizontal and vertical)
- the Easy Edit "Add section" function.

If the "Add menu entry" icon in the Quick-Edit bar is selected, the menu level and page are created in a two-stage dialog. The current step is now visualised at the top edge of the dialog:


Add menu entry - Mozilla Firefox
webedit/inplace/new_menu.jsp?selectedNodeId=98599
Add menu entry
Add new page

In which position should the new menu level appear in Navigation?

Below menu level: Startpage

at position: At last position

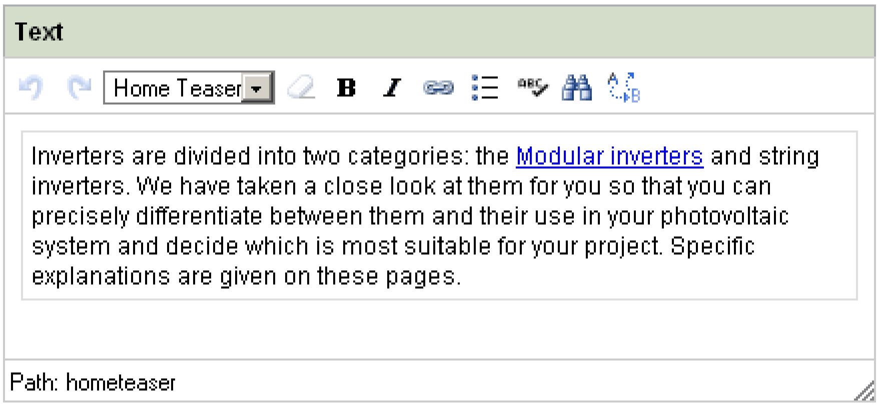
Navigation: Name

Language	Menu name	Keywords	Comment	Navigation menu	Site map
DE				☑	☑
EN	Menu item			☑	☑

Figure 4-28: "Create Menu Level" dialog

4.3.7 Revised DOM Editor

Due to the new compatibility with Mozilla Firefox Version 3.x introduced in 4.2R4, for technical reasons the design of the DOM Editor input component also had to be changed slightly. This primarily affects the icons, but their function has basically remained unchanged:



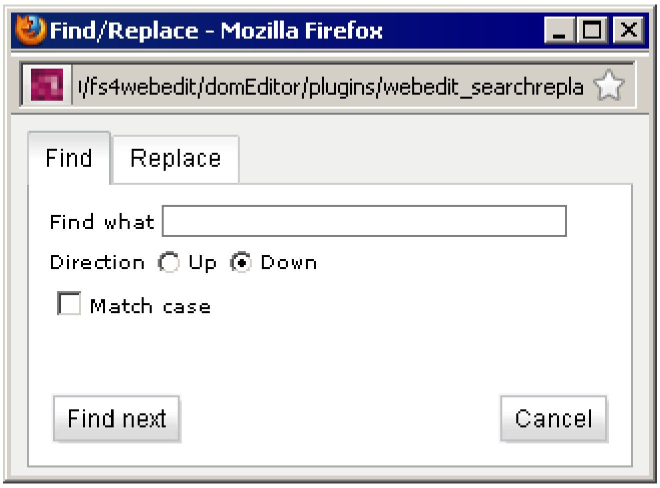
Inverters are divided into two categories: the Modular inverters and string inverters. We have taken a close look at them for you so that you can precisely differentiate between them and their use in your photovoltaic system and decide which is most suitable for your project. Specific explanations are given on these pages.

Path: hometeaser

Figure 4-29: Revised DOM Editor

On the other hand, a new feature is that now **list items** can be indented using the context menu entry "Indent list item" (increase list item indent), so that nested lists can be generated.

In addition, the new **Find/Replace function** can be used to search through the text in the respective DOM Editor and if necessary replace it. Click the 🔍 icon and the dialog window for a full text search is opened:



l/fs4webedit/domEditor/plugins/webedit_searchrepla

Find what

Direction ○ Up ● Down

□ Match case

Figure 4-30: Search dialog

The text of the DOM Editor is always only searched in the direction given in the search direction option. If the searched for text is to be replaced with another text, switch to the "Replace" tab. Select the icon to open the same dialog window as for . However, in this case the "Replace" tab is displayed initially.

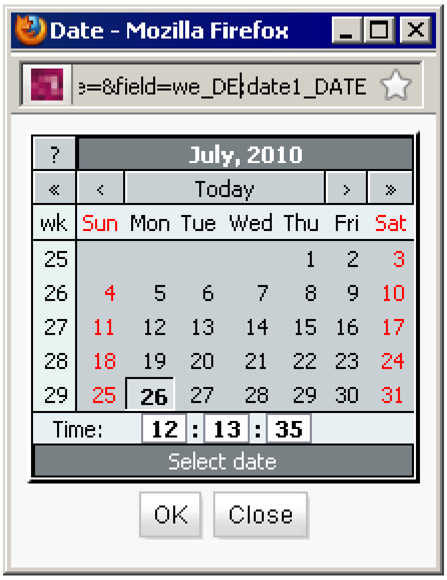
When using the spell checking module in the DOM Editor, the **Ignore word** function is now available on unknown words (marked red) in the context menu. If this function is selected, the spell check is only ignored for current occurrence of the word in the input component, the red marking is removed. Other occurrences of the same (unknown) word in the edit continue to be marked red. The marked word is not added to the dictionary, it is marked again the next time the spell check is activated.

Several functions performed previously using icons or keyboard short cuts can now be reached via a **context menu**:

- cut, copy and paste selected text
- indent or outdent list item, edit list
- edit or delete link

4.3.8 Selection of date / time with specification of seconds via calendar view

From Version 4.2R4 seconds can be entered now also in the WebClient in the input component for selecting date and time (CMS_INPUT_DATE) via the calendar view, opening by means of the icon :



wk	Sun	Mon	Tue	Wed	Thu	Fri	Sat
25					1	2	3
26	4	5	6	7	8	9	10
27	11	12	13	14	15	16	17
28	18	19	20	21	22	23	24
29	25	26	27	28	29	30	31

Time: 12 : 13 : 35

Figure 4-31: Calendar view with field for seconds

4.3.9 WebEdit Help changeover

In many places in the WebEdit interface, a Help button or <F1> can be used to open the context-sensitive WebEdit Help. With Version 4.2R4, Help no longer opens in a separate window as to date, instead the *FirstSpirit Manual for Editors (WebEdit)* is opened in PDF format, always in the latest version and in the language selected on the FirstSpirit start page. The relevant chapter is displayed, depending on the context in which Help is opened.

5 New/changed Functions for Template Developers

In Version 4.2R4, the FirstSpirit component and module model has been extensively revised ("refactoring") and made more flexible. Among other things, the technical basis of all input components has been replaced. In this context, the *FirstSpirit Manual for Developers (Components)* (only available in German) has been revised. Furthermore, the FirstSpirit AppCenter (see Chapter 3 page 16) permits now the development and integration of individual applications for the JavaClient.

Apart from access to the familiar FirstSpirit Access-API, within this scope, access to the so-called Developer-API and the application integration API ("Application-API") is now also enabled. The difference between the Access and Developer-API mainly results from different stability conditions: The **Access-API** is subjected to higher stability conditions, among other things, it is stable within a major version line; an announcement is made beforehand in a minor version before changes are made to methods, classes and functions (d*eprecation*). In the **Developer-API**, on the other hand, changes can be made with a minor version change. The two API documentations can be identified by different colours: Access-API = blue, Developer-API = blackberry. The **Application-API** on the other hand is the most important entry point to application integration. This API can be used to control applications tabs in the application area of the FirstSpirit JavaClient. The Application-API is part of the Developer-API.

See also FirstSpirit Online Documentation, section "Template development" / "FirstSpirit API" and Chapter 5.13 from page 107.

With FirstSpirit Version 4.2 a fundamental revision and consolidation of FirstSpirit's model of input components has begun. This has been finished to a large extent in 4.2R4. The input components with the prefix *FS_* which have been introduced newly in this context have resumed functions of already existing input components. Advantages of these new input components are that often only one input component can be used for diverse use cases. Thus, the number of different page and section templates in one project can be reduced. For example, if the editor should be enabled to select pictures **and** PDF files, only FS_REFERENCE is now required instead of two input components CMS_INPUT_PICTURE and CMS_INPUT_FILE.

If CMS_INPUT_PICTURE and / or CMS_INPUT_FILE are to be replaced by FS_REFERENCE in a project, the respective forms and output channels must be migrated. As Chapter 5.1 page 74 will show, the input component FS_REFERENCE can be configured, if required, in that way that it will offer (only) the same functions as CMS_INPUT_PICTURE or CMS_INPUT_FILE. But if you want to use

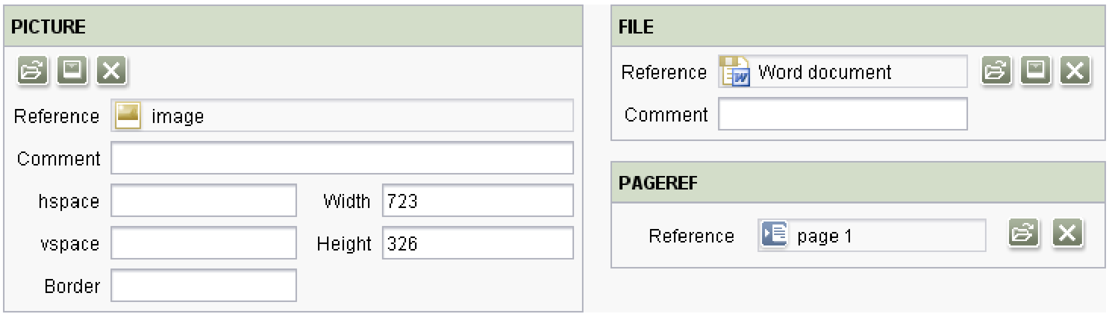
FS_REFERENCE, take into account all new options which are provided by this input component: Can you integrate, for example, two templates with a CMS_INPUT_PICTURE and a CMS_INPUT_FILE form into one template with one FS_REFERENCE form? Another advantage is that the syntax has been consolidated and thus, it has become more logical.

The following chapter describes the steps which are necessary to switch a project to the use of FS_DATASET and / or FS_REFERENCE. First, the default configuration of the components will be shown, then the principal differences regarding function and which parameters have to be adjusted.

5.1 Input components: Migration to FS_REFERENCE

For FirstSpirit Version 5.0, as announced in Version 4.2, it is planned that the input components

- CMS_INPUT_PICTURE
- CMS_INPUT_FILE
- CMS_INPUT_PAGEREF



PICTURE

Reference: image

Comment:

hspace:

Width: 723

vspace:

Height: 326

Border:

FILE

Reference: Word document

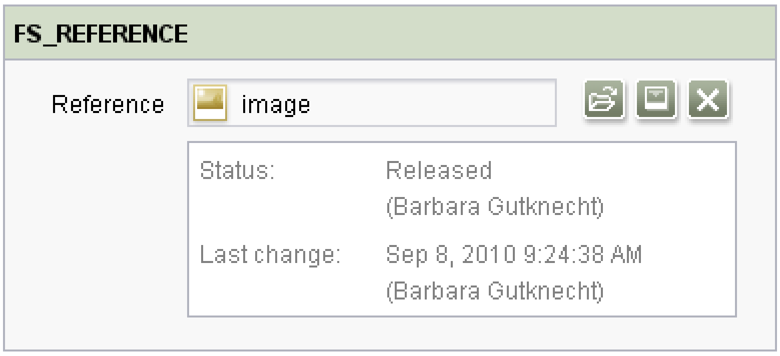
Comment:

PAGEREF

Reference: page 1

Figure 5-1: CMS_INPUT components – default configuration

will be removed and replaced by FS_REFERENCE



FS_REFERENCE

Reference: image

Status: Released (Barbara Gutknecht)

Last change: Sep 8, 2010 9:24:38 AM (Barbara Gutknecht)

Figure 5-2: FS_REFERENCE – default configuration

The following chapters will show what you should notice when switching from a CMS_INPUT_ component to FS_REFERENCE. The examples refer only to the case in which FS_REFERENCE should offer exactly the same functions as the respective CMS_INPUT_ component. But in addition, FS_REFERENCE offers many other options.

5.1.1 Functional differences between FS_REFERENCE and the respective CMS_INPUT_ components

The input components CMS_INPUT_PICTURE, CMS_INPUT_FILE and CMS_INPUT_PAGEREF allow by default the selection of one FirstSpirit object type (pictures, files or page references). The respective selection dialog shows by default only that store from which a reference can be selected, e.g. the Media Store in the case of CMS_INPUT_PICTURE and CMS_INPUT_FILE and the Site Store in the case of CMS_INPUT_PAGEREF. FS_REFERENCE in contrast allows the selection of any object type by default, the selection dialog shows all Stores with all existing folders. If you would like to reduce the **selection** in FS_REFERENCE to special object types you must use the tags `FILTER` / `ALLOW`. If you would like to restrict the **display in the selection dialog** to special object types, stores and / or folders you must use the tags `FILTER` / `HIDE` and the tags `SOURCES` / `FOLDER`. For more detailed information about these tags see also *FirstSpirit Online Documentation*, area "Template development" / "Forms" / "Input components (new)" / "FS_REFERENCE".

Example default configuration CMS_INPUT_FILE:

```
<CMS_INPUT_FILE name="st_file">
      <LANGINFOS>
        <LANGINFO lang="*" label="Selection of a file"/>
      </LANGINFOS>
</CMS_INPUT_FILE>
```

Only files of the Media Store will be shown in the middle column of the selection dialog, but no folders.

The reproduction by FS_REFERENCE looks like this:

```
<FS_REFERENCE name="st_file">
      <FILTER>
        <ALLOW type="file"/>
        <HIDE type="picture"/>
        <HIDE type="mediafolder"/>
      </FILTER>
      <LANGINFOS>
        <LANGINFO lang="*" label="Selection of a file"/>
      </LANGINFOS>
      <PROJECTS>
        <LOCAL name=".">
          <SOURCES>
            <FOLDER name="root" store="mediastore"/>
          </SOURCES>
        </LOCAL>
      </PROJECTS>
```

```
</FS_REFERENCE>
```

Using `SOURCES` / `FOLDER` the display is restricted to the Media Store (`store="mediastore"`), using `FILTER` / `HIDE` the display is restricted to files by hiding pictures and Media folders (`type="picture"` and `type="mediafolder"`). Using `FILTER` / `ALLOW` the selection is restricted to files (`type="file"`).

Example default configuration CMS_INPUT_PAGEREF:

```
<CMS_INPUT_PAGEREF name="st_pageref" hFill="yes">
      <LANGINFOS>
        <LANGINFO lang="*" label="Selection of a page
                                                  reference"/>
      </LANGINFOS>
</CMS_INPUT_PAGEREF>
```

Menu levels, page references and document groups of the Site Store will be shown in the middle column of the selection dialog, but only page references and document groups can be selected.

The reproduction by FS_REFERENCE looks like this:

```
<FS_REFERENCE name="st_file" hFill="yes">
      <FILTER>
        <ALLOW type="pageref"/>
        <ALLOW type="documentgroup"/>
      </FILTER>
      <LANGINFOS>
        <LANGINFO lang="*" label=" Selection of a page
                                                   reference"/>
      </LANGINFOS>
      <PROJECTS>
        <LOCAL name=".">
          <SOURCES>
            <FOLDER name="root" store="sitestore"/>
          </SOURCES>
        </LOCAL>
      </PROJECTS>
</FS_REFERENCE>
```

Using `SOURCES` / `FOLDER` the display is restricted to the Site Store (`store="sitestore"`), using `FILTER` / `ALLOW` the selection is restricted to page references (`type="pageref"`) and document groups (`type="documentgroup"`).

The switch of the forms described in the following chapters will base principally on these default configurations.

5.1.2 Migration of existing forms

From Version 4.2R4, values which have been entered by means of the CMS_INPUT_PICTURE, CMS_INPUT_FILE and CMS_INPUT_PAGEREF input components are imported and processed by FS_REFERENCE. The changeover from CMS_INPUT_ to FS_REFERENCE is possible with the help of a few manual changes in the form.

For the migration, to this end, the `CMS_INPUT_` tag must be replaced by the `FS_REFERENCE` in the syntax. The identifier of the input component given via the `name` parameter, on the other hand, must remain. Only then it can be ensured that data which have been entered by an editor already will be saved and remain editable furthermore.

Example:

```
<CMS_INPUT_PICTURE name="st_picture"...>
```

must be replaced with

```
<FS_REFERENCE name="st_picture"...>
```

Apart from some exceptions (see Chapter 5.1.3 page 77), the other tags and parameters contained in the CMS_INPUT_ input components form are also valid for FS_REFERENCE and can be adopted without change.

5.1.3 Exceptions

Some parameters of the CMS_INPUT_ input components are no more supported by FS_REFERENCE. But only in few cases a corresponding reproduction by FS_REFERENCE is not possible. This chapter shows which parameters must be replaced and how to retain the functionality of the CMS_INPUT_ input components in FS_REFERENCE. However, if the variable name, defined by using `name` (see Chapter 5.1.2 page 77), is retained, also differing functions can be configured for FS_REFERENCE.

- **CMS_INPUT_PICTURE:** The `lean` parameter, with which the display of the input component is controlled, is not supported by FS_REFERENCE and, for a successful migration, must be removed from the form without replacement.
- **CMS_INPUT_FILE:** The `allowFolder` parameter, which enables the selection of folders, is not supported by FS_REFERENCE. The parameter must be removed from the form for successful migration. If `allowFolder="yes"` was

previously set in the CMS_INPUT_FILE input component, the following syntax must be changed, e.g.

```
<CMS_INPUT_FILE name="st_file" allowFolder="yes">
      <LANGINFOS>
        <LANGINFO lang="*" label="Selection of a file"/>
      </LANGINFOS>
</CMS_INPUT_FILE>
```

```
<FS_REFERENCE name="st_file">
      <FILTER>
        <ALLOW type="file"/>
        <HIDE type="picture"/>
        <ALLOW type="mediafolder"/>
      </FILTER>
      <LANGINFOS>
        <LANGINFO lang="*" label="Selection of a file"/>
      </LANGINFOS>
      <PROJECTS>
        <LOCAL name=".">
          <SOURCES>
            <FOLDER name="root" store="mediastore"/>
          </SOURCES>
        </LOCAL>
      </PROJECTS>
</FS_REFERENCE>
```

Because of rare use, FS_REFERENCE does no more provide some fields which existed in the CMS_INPUT_ input components. For example: fields for entering a comment, horizontal and vertical space, height and width of a picture at CMS_INPUT_PICTURE and CMS_INPUT_FILE. If these fields have been used in a project this must be taken into account when switching to FS_REFERENCE.

- **CMS_INPUT_PAGEREF:** The `mediaStore` parameter, which enables the selection of media from the Media Store, is not supported by FS_REFERENCE. The parameter must be removed from the form for successful migration. If `mediastore="yes"` was set in the CMS_INPUT_PAGEREF input component, the following syntax must be added to the new FS_REFERENCE form to be used:

```
<CMS_INPUT_PAGEREF name="st_pageref" hFill="yes" mediastore="yes">
      <LANGINFOS>
        <LANGINFO lang="*" label="Selection of a page reference"/>
      </LANGINFOS>
</CMS_INPUT_PAGEREF>
```

```
<FS_REFERENCE name="st_file" hFill="yes">
      <FILTER>
        <ALLOW type="pageref"/>
        <ALLOW type="documentgroup"/>
        <ALLOW type="media"/>
      </FILTER>
      <LANGINFOS>
        <LANGINFO lang="*" label="Selection of a page reference"/>
      </LANGINFOS>
      <PROJECTS>
        <LOCAL name=".">
          <SOURCES>
            <FOLDER name="root" store="sitestore"/>
            <FOLDER name="root" store="mediastore"/>
          </SOURCES>
        </LOCAL>
      </PROJECTS>
</FS_REFERENCE>
```

The `sections` parameter, which can be used to specify in CMS_INPUT_PAGEREF whether only page references, only sections or page references **and** sections can be selected in the input component, is also available for FS_REFERENCE. In comparison to CMS_INPUT_PAGEREF the values have changed:

	CMS_INPUT_PAGEREF	FS_REFERENCE
page only	`page` (default)	`no`
page and sections	`both`	`yes` (default)
sections only	`section`	`only`

If `sections="page"` was set in the CMS_INPUT_PAGEREF input component or no value was set for `sections`, `sections="no"` must be set instead in the new FS_REFERENCE form to be used. If `sections="section"` was set in the CMS_INPUT_PAGEREF input component, `sections="only"` must be set instead in the new FS_REFERENCE form to be used:

```
<CMS_INPUT_PAGEREF name="st_pageref" hFill="yes"
                      sections="section">
      <LANGINFOS>
        <LANGINFO lang="*" label="Selection of a page reference"/>
      </LANGINFOS>
</CMS_INPUT_PAGEREF>
```

```
<FS_REFERENCE name="st_file" hFill="yes" sections="only">
      <FILTER>
        <ALLOW type="pageref"/>
        <ALLOW type="documentgroup"/>
      </FILTER>
      <LANGINFOS>
        <LANGINFO lang="*" label="Selection of a page reference"/>
      </LANGINFOS>
      <PROJECTS>
        <LOCAL name=".">
          <SOURCES>
            <FOLDER name="root" store="sitestore"/>
          </SOURCES>
        </LOCAL>
      </PROJECTS>
</FS_REFERENCE>
```

If `sections="both"` was set in the CMS_INPUT_PAGEREF input component, `sections="yes"` must be set instead in the new FS_REFERENCE form to be used.

The `sortOrder` parameter, with which the sort order of the sections in the selection dialog can be influenced, is not supported by FS_REFERENCE. The parameter must be removed from the form for successful migration. In FS_REFERENCE, selectable sections are always displayed in the same order as they exist in the tree structure.

5.1.4 Migration of the output channels

If you would like to switch from the input components CMS_INPUT_PICTURE, CMS_INPUT_FILE and CMS_INPUT_PAGEREF to FS_REFERENCE, the content of the output channel tabs (instructions and methods) must be adapted as the case may be, too. Because the CMS_INPUT_ input components deliver objects of another data type than FS_REFERENCE:

CMS_INPUT_PICTURE: Data type *GraphicalMedium*	FS_REFERENCE: Data type *TargetReference*
CMS_INPUT_FILE: Data type *BinaryMedium*	
CMS_INPUT_PAGEREF: Data type *ElementReference*	

For this reason, you should check if methods, functions and instructions which can be used with objects of the data types *GraphicalMedium, BinaryMedium* and *ElementReference* can be used also on an object of the data type *TargetReference*. The identifier of the input component which is defined in the form by the parameter `name` must be resumed in any case.

In the following examples the syntax of the HTML channel can be resumed:

- `getClass`:
    - o Data type *GraphicalMedium*: `$CMS_VALUE(pt_picture.class)$`
    - o Data type *BinaryMedium*: `$CMS_VALUE(pt_file.class)$`
    - o Data type *ElementReference*:`$CMS_VALUE(pt_pageref.class)$`
    - o Data type *TargetReference:* `$CMS_VALUE(pt_reference.class)$`
- `$CMS_REF(...)$`:
    - o Data type *GraphicalMedium*: `$CMS_REF(pt_picture)$`
    - o Data type *BinaryMedium*: `$CMS_REF(pt_file)$`
    - o Data type *ElementReference*:`$CMS_REF(pt_pageref)$`
    - o Data type *TargetReference:* `$CMS_REF(pt_reference)$`
- *ref(...)* function and `getUrl`:
    - o Data type *GraphicalMedium*: `$CMS_VALUE(ref(pt_picture).url)$`
    - o Data type *BinaryMedium*: `$CMS_VALUE(ref(pt_file).url)$`

  - Data type *ElementReference*: `$CMS_VALUE(ref(pt_pageref).url)$`
  - Data type *TargetReference:* `$CMS_VALUE(ref(pt_reference).url)$`
- `getSectionName`:
  - Data type *ElementReference*: `$CMS_VALUE(pt_pageref.sectionName)$`
  - Data type *TargetReference:* `$CMS_VALUE(pt_reference.sectionName)$`

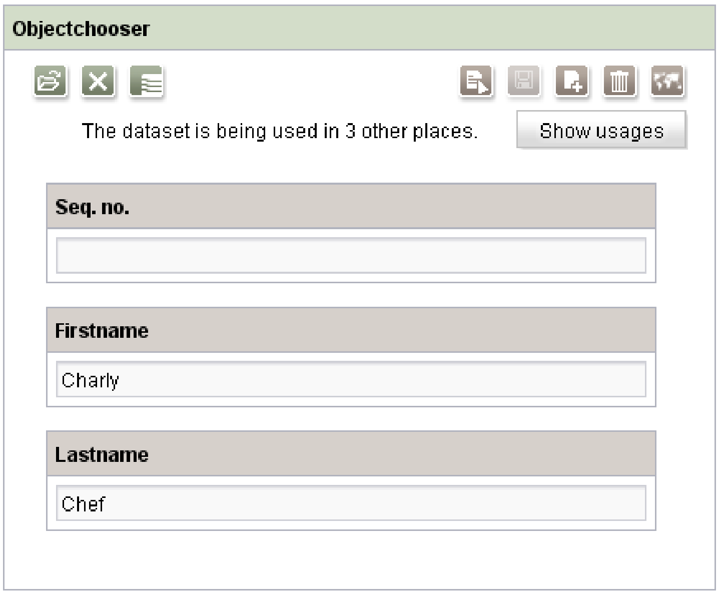
For example, the output must be adapted in the following cases:

- `getUid`:
  - Data type *GraphicalMedium:* `$CMS_VALUE(pt_picture.medium.uid)$`
  - Data type *BinaryMedium*: `$CMS_VALUE(pt_file.medium.uid)$`
  - Data type *ElementReference*: `$CMS_VALUE(pt_pageref.pageRef.uid)$`
  - Data type *TargetReference:* `$CMS_VALUE(pt_reference.uid)$`
- `getRemoteName` / `getRemote`:
  - Data type *GraphicalMedium:* `$CMS_VALUE(pt_picture.remoteName)$`
  - Data type *BinaryMedium*: `$CMS_VALUE(pt_file.remoteName)$`
  - Data type *ElementReference:*`$CMS_VALUE(pt_pageref.remote)$`
  - Data type *TargetReference*: `$CMS_VALUE(pt_reference.remoteSymName)$`
- *ref(...)* function and `getRemoteName` / `getRemote`:
  - Data type *GraphicalMedium*: `$CMS_REF(pt_file, remote:pt_file.remoteName)$`
  - Data type *BinaryMedium*: `$CMS_REF(pt_picture, remote:pt_picture.remoteName)$`
  - Data type *ElementReference*: `$CMS_REF(pt_pageref, remote:pt_pageref.remote)$`
  - Data type *TargetReference:* `$CMS_REF(pt_reference, remote:pt_reference.remoteSymName)$`

> ! *For all other usages you must check if methods, instructions and functions must be adapted. See also FirstSpirit Online Documentation.*

5.2 Input components: Migration to FS_DATASET

For FirstSpirit Version 5.0, as announced in Version 4.2, it is planned that the input component CMS_INPUT_OBJECTCHOOSER



Objectchooser

The dataset is being used in 3 other places.

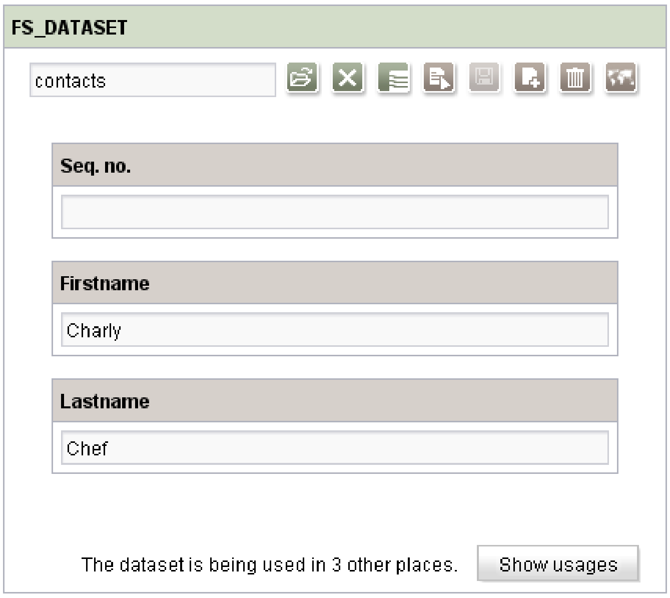
Seq. no.

Firstname: Charly

Lastname: Chef

Figure 5-3: CMS_INPUT_OBJECTCHOOSER – default configuration

will be removed and replaced by FS_DATASET:



FS_DATASET

contacts

Seq. no.

Firstname: Charly

Lastname: Chef

The dataset is being used in 3 other places.

Figure 5-4: FS_DATASET – default configuration

5.2.1 Functional differences between FS_DATASET and CMS_INPUT_OBJECTCHOOSER

Unlike CMS_INPUT_OBJECTCHOOSER, FS_DATASET is by default not limited to the selection of data records from a defined target table. Instead, the desired table can be selected using a selection dialog. At the same time, the target table and desired data record are selected.

5.2.2 Migration of existing forms

From Version 4.2R4, values which have been entered by means of the CMS_INPUT_OBJECTCHOOSER input component are imported and processed by FS_DATASET. The changeover of the form from CMS_INPUT_OBJECTCHOOSER to FS_DATASET is possible with the help of a few manual changes.

For the migration, to this end, the `CMS_INPUT_OBJECTCHOOSER` tag must be replaced by the `FS_DATASET` tag. The identifier of the input component given via the `name` parameter, on the other hand, must remain. Only then it can be ensured that data which have been entered by an editor already will be saved and remain editable furthermore.

Example:

```
<CMS_INPUT_OBJECTCHOOSER name="st_dataset"...>
```

must be replaced with

```
<FS_DATASET name="st_dataset"...>
```

If the selection should be restricted to the table which was defined for CMS_INPUT_OBJECTCHOOSER, the reference to the table template which was specified for CMS_INPUT_OBJECTCHOOSER using the mandatory parameter `table`, must be specified in FS_DATASET using the parameter `name` within the tags `SOURCES` and `CONTENT`. Unlike as it was the case for CMS_INPUT_OBJECTCHOOSER, in FS_DATASET the reference name of the desired data source must be specified, e.g.

```
<SOURCES>
     <CONTENT name="produkte"/>
</SOURCES>
```

In addition, the unique identifier of the table template from the CMS_INPUT_OBJECTCHOOSER form which is to be migrated, is needed for that the data which have been entered by the editor by means of CMS_INPUT_OBJECTCHOOSER already are saved and can be edited furthermore. The identifier of the template must be given with the tag `LEGACYTEMPLATE`, e.g.
`<LEGACYTEMPLATE>Products.products</LEGACYTEMPLATE>`

5.2.3 Migration of the output channels

If you would like to switch from the input component CMS_INPUT_OBJECTCHOOSER to FS_DATASET, the content of the output channel tabs (instructions and methods) must be adapted as the case may be, too. Because CMS_INPUT_OBJECTCHOOSER delivers an object of another data type than FS_DATASET:

CMS_INPUT_OBJECTCHOOSER:	FS_ DATASET:
Data type *Entity*	Data type *DatasetContainer*

For this reason, you should check if methods and instructions which can be used with objects of the data type *Entity* can be used also on objects of the data type *DatasetContainer.* The identifier of the input component which is defined in the form by the parameter `name` must be resumed in any case.

While *get* methods which are called on the data type object *Entity* are executed directly on a data record, the data type object *DatasetContainer* contains in addition to the data record also the related table template. For this reason, the method `.getDataset().getEntity()` (in Bean syntax: `.dataset.entity`) must be added when switching from CMS_INPUT_OBJECTCHOOSER to FS_DATASET.

Example:

I.e., the output of the value of "column1" of a data record, selected by means of CMS_INPUT_OBJECTCHOOSER, using

```
$CMS_VALUE(st_dataset.column1)$
```

should be

```
$CMS_VALUE(st_dataset.dataset.entity.column1)$
```

when switching to FS_DATASET.

For all other usages you must check if methods, instructions and functions must be adapted. See also FirstSpirit Online Documentation.

5.3 Input components: Further development of FS_LIST

Since its introduction in the initial release version of 4.2, the input component FS_LIST has shown the functions of the input component CMS_INPUT_CONTENTAREALIST. The objective of the consolidation of the FirstSpirit input components started in 4.2 should be to group together all input components with a list character in FS_LIST. Therefore, in Version 4.2R4 the functions of the following input components have been added:

- CMS_INPUT_CONTENTLIST
- CMS_INPUT_LINKLIST
- CMS_INPUT_SECTIONLIST
- CMS_INPUT_TABLIST

To this end, FS_LIST can access data of different origin or different types. The data type is defined using the parameter `type` in the `DATASOURCE` tag. Depending on the data type, other tags can be used to configure the input component. For example, the template developer can also specify which icons are to be available to the editor for including and editing list contents (e.g. "Add", "Delete", "Edit").

The FirstSpirit Online Documentation provides an example project which illustrates exemplary configurations of FS_LIST and the migration from the CMS_INPUT_ input components to FS_LIST (form and output). See "Template development" / "Forms" / "Input components (new)" / "FS_LIST" / "Sample project".

! *In **WebEdit**, FS_LIST in 4.2R4 is only supported in a rudimentary way; at present, only the functions of the input component CMS_INPUT_CONTENTAREALIST (list for managing section contents) are available. To provide more functions, the so called (not released!) applet mode can be activated for FS_LIST (Project settings, "WebEdit settings", activate option "Use editor Applet", enter "FS_LIST" into the field "Applet editors"). However, this will only be a transition solution; native FS_LIST support will exist from FirstSpirit Version 5.0.*

5.3.1 Configuration

The tags and parameters which can be used for configuring FS_LIST within the form area is described in details in the *FirstSpirit Online Documentation*, area "Template development" / "Forms" / "Input components (new)" / "FS_LIST".

! *FS_LIST only supports the selection of links of the type "genericLink" for the DATASOURCE type INLINE with `source="LINKTEMPLATES"`! For this reason it is recommended that links of the types "internalLink", "externalLink" and "contentLink" are converted **before** FS_LIST is used in a project. For this purpose, the conversion function can be used (context menu on link templates, "Extras" / "Convert link template").*

! *In **WebEdit,** FS_LIST is only supported for the DATASOURCE type INLINE with `source="SECTIONTEMPLATES"` (corresponds to CMS_INPUT_CONTENTAREALIST).*

5.3.2 Migration of existing forms

The migration of the input components to be replaced by FS_LIST (CMS_INPUT_CONTENTAREALIST, CMS_INPUT_CONTENTLIST, CMS_INPUT_LINKLIST, CMS_INPUT_SECTIONLIST and CMS_INPUT_TABLIST), to FS_LIST is possible by adjusting the forms.

The syntax of FS_LIST has been changed for Version 4.2R4. For this reason, usages of FS_LIST which have been created with a FirstSpirit Version below 4.2R4 ***must*** *be adjusted. For the switch from the "old" FS_LIST to the new FS_LIST the same notes do apply as for the switch from CMS_INPUT_CONTENTAREALIST to the new FS_LIST.*

For the migration, to this end, the `CMS_INPUT_` tag must be replaced by the `FS_LIST` tag in the syntax, e.g. `<CMS_INPUT_LINKLIST ...>` must be replaced by `<FS_LIST ...>` in the form. The identifier of the input component given via the `name` parameter, on the other hand, must remain. Many of the parameters used in the CMS_INPUT_ input components can be adopted for FS_LIST; on the other hand, the following tags and parameters must be replaced or added:

For all input components:

- The parameters `maxEntries` which is used for defining how many entries can be selected or added maximally, and `useLanguages` which enables or disables multilingual maintenance of data, have been specified within the `CMS_INPUT_` tags in the CMS_INPUT_ input components. These parameters can be assumed in FS_LIST, however they must be used now in the `DATASOURCE` tag, e.g.

```
<CMS_INPUT_CONTENTLIST name="st_datasets"
         table="Products.contacts" maxEntries="5"
         useLanguages="no" ...>
```

becomes

```
<FS_LIST name="st_sections" ...>
         <DATASOURCE type="inline" maxEntries="5
                    useLanguages="no">
         <TABLE>Products.contacts</TABLE>
         ...
```

- As a default, FS_LIST input components are 500 pixels high and therefore probably higher than the previously used CMS_INPUT input components. If the height is to be reduced, the relevant `height` parameter must be started with the required number of pixels, e.g.

```
<FS_LIST name="st_sections" height="300" .../>
```

5.3.2.1 Replacements for CMS_INPUT_CONTENTAREALIST and FS_LIST

- `<DATASOURCE type="INLINE">` must be added.
- The `SOURCES` tag must be replaced by the `TEMPLATES` tag, in addition, `source="sectiontemplates"` must be added.
- If selectable section templates were defined in the input component using `SOURCES` / `TEMPLATE`, the `TEMPLATE` tag must be retained, the parameter `name` must be replaced by the parameter `uid`, however, the value remains.
- The following syntax must be added to enable editing of the list entries as in CMS_INPUT_CONTENTAREALIST:

```
<ACTIONS>
          <ACTION name="ADD"/>
          <ACTION name="REMOVE"/>
          <ACTION name="UP"/>
          <ACTION name="DOWN"/>
          <ACTION name="EDIT"/>
</ACTIONS>
```

- If a labelling for the sections of the list was defined by means of the tags `VARIABLES` / `VARIABLE`, this must be reproduced by means of the tags `LABELS` / `LABEL` in FS_LIST. Using `#item` you can access the value which is stored in an input component of the section. For this purpose, the variable (identifier) of the respective input component must be attached, separated by a dot, e.g.

```
<LABELS>
          <LABEL lang="*">
                      #item.VARIABLE
          </LABEL>
</LABELS>
```

- If the `template` variable, with which the reference name of the template used for an entry was output, was defined within the `VARIABLES` / `VARIABLE` tags in CMS_INPUT_CONTENTAREALIST, the system object `#template` for this is now used within the `LABELS` / `LABEL` tags in FS_LIST. The definition

```
<VARIABLES>
          <VARIABLE name="template"/>
</VARIABLES>
```

then became in FS_LIST

```
<LABELS>
        <LABEL lang="*">#template.uid</LABEL>
</LABELS>
```

- To get the layout of CMS_INPUT_CONTENTAREALIST the following syntax can be added:

```
<COLUMNS>
            <COLUMN show="no">#identifier</COLUMN>
</COLUMNS>
<LAYOUT>
            <ADD component="toolbar" constraint="top"/>
            <ADD component="overview" constraint="center"/>
            <ADD component="stackedview" constraint="hide"/>
</LAYOUT>
```

There is no counterpart in FS_LIST for the function in CMS_INPUT_CONTENTAREALIST for manually assigning a name or renaming sections (e.g. aI icon), which is then displayed in the overview. If the selected section only contains "empty" input components (e.g. when a new one is created, or if not values have been entered yet), as a default the display name of the section template is displayed for the list overview and on the bar, if input components are "filled" the lettering and content of the input components are displayed.

In order for section names already assigned manually to be retained and kept saved in FS_LIST, an additional input component must be defined with the variables (identifiers) `displayName` *(case sensitive – note upper and lower cases!), in which this section name is to be saved in the section template(s) concerned (selected using CMS_INPUT_CONTENTAREALIST). The input component must be of the type CMS_INPUT_TEXT or CMS_INPUT_TEXTAREA.*

A corresponding FS_LIST form could be in this case like this:

```
<FS_LIST name="st_cal">
           <DATASOURCE type="inline" useLanguages="no">
              <LABELS>
                 <LABEL lang="*">#item.displayName</LABEL>
              </LABELS>
              <ACTIONS>
                 <ACTION name="ADD">
                     <PARAM name="select-ui">popup</PARAM>
                 </ACTION>
                 <ACTION name="REMOVE"/>
                 <ACTION name="UP"/>
                 <ACTION name="DOWN"/>
                 <ACTION name="EDIT"/>
             </ACTIONS>
              <COLUMNS>
                 <COLUMN show="no">#identifier</COLUMN>
              </COLUMNS>
              <LAYOUT>
                 <ADD component="toolbar" constraint="top"/>
                 <ADD component="overview" constraint="center"/>
                 <ADD component="stackedview" constraint="hide">
                 <PARAM name="show-language-tabs">yes</PARAM>
                 </ADD>
              </LAYOUT>
              <TEMPLATES source="sectiontemplates"/>
          </DATASOURCE>
          <LANGINFOS>
              <LANGINFO lang="*" label="Absätze"/>
          </LANGINFOS>
</FS_LIST>
```

When using CMS_INPUT_CONTENTAREALIST, a list of sections could be created at the top of a page, for example for a form of an index of contents, using the following syntax:

```
$CMS_FOR(entry, st_cal)$
    <a href="#$CMS_VALUE(entry.name)$">Section
$CMS_VALUE(#for.index + 1)$</a>
$CMS_END_FOR$
```

When using FS_LIST, the syntax should be adapted like this:

```
$CMS_FOR(entry, st_cal)$
    <a href="#$CMS_VALUE(if(entry.displayName.isEmpty, "st_cal_" +
#for.index, entry.displayName))$">Section $CMS_VALUE(#for.index +
1)$</a>
$CMS_END_FOR$
```

The following syntax should be added in the section template, which is selected by the editor, as link target (anchor):

```
$CMS_SET(anchorName, if(displayName.isEmpty, "st_cal_" + #index,
displayName))$
   <a id="$CMS_VALUE(anchorName)$"
name="$CMS_VALUE(anchorName)$"></a>
```

5.3.2.2 Replacements for CMS_INPUT_LINKLIST

- `<DATASOURCE type="INLINE"` must be added.
- The `LINKEDITORS` tag must be replaced by the `TEMPLATES` tag, in addition, `source="linktemplates"` must be added.
- If selectable link templates were defined in the input component via `LINKEDITORS` / `LINKEDITOR` the `LINKEDITOR` tag must be replaced by `TEMPLATE`, the parameter `name` must be replaced by the parameter `uid`, however, the value is left as it is.
- The following syntax must be added to enable editing of the list entries as in CMS_INPUT_LINKLIST:

```
<ACTIONS>
          <ACTION name="ADD"/>
          <ACTION name="REMOVE"/>
          <ACTION name="UP"/>
          <ACTION name="DOWN"/>
          <ACTION name="EDIT"/>
</ACTIONS>
```

- The `LABELS / LABEL` tags are used in FS_LIST to display the value which is entered as link text as labelling text in the list overview and on the bars like in CMS_INPUT_LINKLIST. Using `#item` you can access the value which is stored in an input component of the link template. For this purpose, the variable (identifier) of the respective input component must be attached, separated by a dot, e.g.

```
<LABELS>
          <LABEL lang="*">
                      #item.VARIABLE
          </LABEL>
</LABELS>
```

- To get the layout of CMS_INPUT_LINKLIST the following syntax can be added:

```
<COLUMNS>
            <COLUMN show="no">#identifier</COLUMN>
</COLUMNS>
<LAYOUT>
            <ADD component="toolbar" constraint="top"/>
            <ADD component="overview" constraint="center"/>
            <ADD component="stackedview" constraint="hide"/>
</LAYOUT>
```

! *FS_LIST only supports the selection of links of the type "genericLink" for the DATASOURCE type INLINE with* `source="LINKTEMPLATES"`*! For this reason it is recommended that links of the types "internalLink", "externalLink" and "contentLink" are converted* ***before*** *FS_LIST is used in a project. For this purpose, the conversion function can be used (context menu on link templates, "Extras" / "Convert link template").*

5.3.2.3 Replacements for CMS_INPUT_CONTENTLIST

- `<DATASOURCE type="DATABASE">` must be added.
- The table template defined using the `table` parameter must be given within an opening and closing `TABLE` tag:

```
<TABLE>Products.contacts</TABLE>
```

- The following syntax must be added to enable editing of the list entries as in CMS_INPUT_CONTENTLIST:

```
<ACTIONS>
            <ACTION name="ADD"/>
            <ACTION name="REMOVE"/>
            <ACTION name="UP"/>
            <ACTION name="DOWN"/>
            <ACTION name="GOTO"/>
            <ACTION name="VIEW"/>
</ACTIONS>
```

- If `allowEdit="yes"` was set in CMS_INPUT_CONTENTLIST, `<ACTION name="EDIT"/>` must be additionally added within the `ACTIONS` tags; if `allowNew="yes"` was set, `<ACTION name="NEW"/>` must be added.
- If in CMS_INPUT_CONTENTLIST the parameter `indexAttribute` was set (for example when using the component as media gallery, tag `MEDIAMODE`), the value which was set for the parameter `indexAttribute` must be given now within an opening and closing `PERSISTENCEINDEX` tag, e.g.
  CMS_INPUT_CONTENTLIST: `indexAttribute="Order_Index"`
  FS_LIST: `<PERSISTENCEINDEX>Order_Index</PERSISTENCEINDEX>`
- While a query for restricting the number of selectable data records was defined by means of the `QUERY` tag **within** the form in CMS_INPUT_CONTENTLIST, it is defined by means of a query in the Template Store in FS_LIST. You must access this query in the FS_LIST form by its reference name as follows:

```
<QUERY name="query_products"/>
```

- To get the layout of CMS_INPUT_CONTENTLIST the following syntax can be added:

```
<COLUMNS>
            <COLUMN show="no">#identifier</COLUMN>
</COLUMNS>
<LAYOUT>
            <ADD component="toolbar" constraint="top"/>
            <ADD component="overview" constraint="center"/>
            <ADD component="stackedview" constraint="hide"/>
</LAYOUT>
```

5.3.2.4 Replacements for CMS_INPUT_TABLIST

- `<DATASOURCE type="DATABASE">` must be added.
- The table template defined using the `table` parameter must be given within an opening and closing `TABLE` tag:

```
<TABLE>Products.contacts</TABLE>
```

- If a flexible labelling of the tabs was defined in CMS_INPUT_TABLIST by means of `tabAttribute` or `expression`, the tags `LABELS` / `LABEL` must be used in FS_LIST. Use `#item` to access the value which is stored in a column. Attach a dot and the variable of the column (identifier of the input component), e.g.

```
<LABELS>
          <LABEL lang="*">
                      #item.VARIABLE
          </LABEL>
</LABELS>
```

- The following syntax must be added to enable editing of the list entries as in CMS_INPUT_TABLIST:

```
<ACTIONS>
            <ACTION name="ADD"/>
            <ACTION name="REMOVE"/>
            <ACTION name="EDIT"/>
</ACTIONS>
```

- If `allowEdit="yes"` was set in CMS_INPUT_TABLIST, add `<ACTION name="EDIT"/>` within the `ACTIONS` tags, if `allowNew="yes"` was set, you must add `<ACTION name="NEW"/>`. If `allowChoose="no"` was set, delete `<ACTION name="ADD"/>`. If `allowEdit="no"` was set, delete `<ACTION name="EDIT"/>`. If `delete="yes"` was set, add `<ACTION name="DELETE"/>`.
- To get the layout of CMS_INPUT_TABLIST the following syntax can be added:

```
<COLUMNS>
            <COLUMN show="no">#identifier</COLUMN>
</COLUMNS>
<LAYOUT>
            <ADD component="toolbar" constraint="top"/>
            <ADD component="overview" constraint="hide"/>
            <ADD component="tabbedview" constraint="bottom"/>
</LAYOUT>
```

5.3.2.5 Replacements for CMS_INPUT_SECTIONLIST

- `<DATASOURCE type="PAGE">` must be added.

Basically, all opened tags must also be closed again accordingly.

5.3.3 Adjusting output channels

If you would like to switch from CMS_INPUT_LINKLIST, CMS_INPUT_CONTENTAREALIST, CMS_INPUT_CONTENTLIST, CMS_INPUT_TABLIST, CMS_INPUT_SECTIONLIST and FS_LIST (created before FirstSpirit Version 4.2R4) to FS_LIST, please check if the output channels must be changed (instructions and methods), because the data type of the objects delivered by the input components are different:

CMS_INPUT_CONTENTAREALIST: Data type *SectionList <Section >*	FS_LIST (from 4.2R4): Data type *FormDataList <IdProvidingFormData>*
CMS_INPUT_CONTENTLIST: Data type *List <Entity >*	
CMS_INPUT_LINKLIST: Data type *List <Link >*	
CMS_INPUT_SECTIONLIST: Data type *List <SectionListEntry >*	
CMS_INPUT_TABLIST: Data type *List <Entity >*	
FS_LIST (bis 4.2R2): Data type *SectionList <Section >*	

! *The status of FS_LIST is "under development" in FirstSpirit Version 4.2. Syntax and API of FS_LIST have been slightly changed compared to 4.2R2, to produce a more consistent syntax. It can not be excluded that the syntax of FS_LIST will change again.*

5.3.3.1 Example CMS_INPUT_CONTENTAREALIST (FS_LIST, Datasource type ***INLINE***)

CMS_INPUT_CONTENTAREALIST returns a list of *Section* objects. A *Section* object represents the content of a section which is created and maintained via CMS_INPUT_CONTENTAREALIST. Using a `$CMS_FOR(…)$` instruction you can access single *Section* objects:

```
$CMS_FOR(_obj, pt_cal)$
    $CMS_VALUE(_obj)$
$CMS_END_FOR$
```

Here, `pt_cal` is the identifier of a CMS_INPUT_CONTENTAREALIST input component, `_obj` one of the sections which are created by CMS_INPUT_CONTENTAREALIST. The data of the single sections are then output by means of the respective section template.

When using the new FS_LIST component, this output syntax can be assumed. Moreover, the data stored in FS_LIST in a single input component can be accessed easily by indicating the identifier of the input component:

```
$CMS_FOR(_obj, pt_cal)$
    $CMS_VALUE(_obj.st_text)$
$CMS_END_FOR$
```

Here, `st_text` is the identifier of the input component in the section template which is selected by means of FS_LIST.

> ! *For information about the use of CMS_INPUT_CONTENTAREALIST in combination with WebEdit "Easy-Edit" see "FirstSpirit Online Documentation", "Template development" / "WebEdit" / "Easy-Edit", section "Special case: CMS_INPUT_CONTENTAREALIST and FS_LIST".*

Data stored in FS_LIST as replacement for CMS_INPUT_LINKLIST is output analogously. However in this case, the identifier of the input component of the selected link template must be used.

5.3.3.2 Example CMS_INPUT_TABLIST (FS_LIST, Datasource type ***DATABASE***)

The input components CMS_INPUT_TABLIST and CMS_INPUT_CONTENTLIST return a list of objects of the data type *Entity* (see also Chapter 5.2.3 page 85). Using a `$CMS_FOR(…)$` instruction you can output the data of the data records which are selected in these input components, e.g. the ID via the method `.getId()`:

```
$CMS_FOR(_obj, pt_tablist)$
    $CMS_VALUE(_obj.getId())$
$CMS_END_FOR$
```

Data of a column of a selected data record can be output by indicating the name of the column:

```
$CMS_FOR(_obj, pt_tablist)$
    $CMS_VALUE(_obj.headline)$
$CMS_END_FOR$
```

Here, `pt_tablist` is the identifier of the CMS_INPUT_TABLIST input component, `_obj` one of the data records selected in CMS_INPUT_TABLIST and `headline` the name of the column (according to the configuration, this can be defined for different languages, too, e.g. `headline_DE` and `headline_EN`).

When using the new FS_LIST component, the variable of the input component which is mapped to the respective table column (tab "Mapping" of the table template) must be indicated instead of the variable of the table column whose data are to be output. The ID can be returned via the key term `fs_id`.

Output of the ID of the selected data records:

```
$CMS_FOR(_obj, pt_tablist)$
    ID of the data record: $CMS_VALUE(_obj.fs_id)$
$CMS_END_FOR$
```

Output of the data of an input component:

```
$CMS_FOR(_obj, pt_tablist)$
    Text: $CMS_VALUE(_obj.cs_headline)$
$CMS_END_FOR$
```

Here, `cs_headline` is the identifier of the input component in the table template of the data record which is selected by means of FS_LIST. For this reason, no language-dependent differentiation is required.

Data stored in FS_LIST as replacement for CMS_INPUT_CONTENTLIST is output analogously.

5.3.3.3 Example CMS_INPUT_SECTIONLIST (FS_LIST, Datasource type ***PAGE***)

The input component CMS_INPUT_SECTIONLIST returns a list of *SectionListEntry* objects. The single section entries are output by means of a `$CMS_FOR(...)$` instruction. The following methods can be applied to an object, for example:

- `getSectionName` (in Bean syntax: `.sectionName`)
  output of the name of the section

- `getAnchorName` (in Bean syntax: `.anchorName`)
  output of the name of the section, special characters are replaced by underscores
  Note: This method may be only relevant for projects which have been created with a FirstSpirit Version below 4.x. From FirstSpirit Version 4.x on, special characters can not be entered in reference names anymore.
- `getText` (in Bean syntax: `.text`)
  output of the (manually assigned) label of the section

Example:

```
<ul>
 $CMS_FOR(_obj, pt_sectionlist)$
    <li>
          <a href="#$CMS_VALUE(_obj.anchorName)$">
           Label: $CMS_VALUE(_obj.text)$      <br>
           UID:   $CMS_VALUE(_obj.sectionName)$</a>
    </li>
 $CMS_END_FOR$
</ul>
```

Here, `pt_sectionlist` is the identifier of a CMS_INPUT_SECTIONLIST input component.

These methods can still be used when migrating to FS_LIST. But if FS_LIST, Datasource type *PAGE*, is introduced newly in a project, the key terms `uid` and `label` should be used instead: `uid` is used to output the name of a section (before: `getSectionName`), `label` is used to output the label (before: `getText`). Since no special characters are allowed in section names from FirstSpirit Version 4.x anymore, `getUid` can be used instead of `getAnchorName` for projects which are created with this version, e.g.:

```
<ul>
 $CMS_FOR(_obj, pt_sectionlist)$
    <li>
          <a href="#$CMS_VALUE(_obj.uid)$">
           Label: $CMS_VALUE(_obj.label)$     <br>
           UID:   $CMS_VALUE(_obj.uid)$</a>
    </li>
 $CMS_END_FOR$
</ul>
```

For all other usages you must check if methods, instructions and functions must be adapted. See also FirstSpirit Online Documentation.

5.4 New input component FS_BUTTON

The input component FS_BUTTON provides the editor with an icon, a button or a link, which is referenced to a FirstSpirit script or a class, which e.g. is provided by a FirstSpirit module on the server. If the editor clicks this control or if they drag objects onto using the mouse cursor ("Drop"), the referenced script or class is run. In this way, FS_BUTTON can be used to create an icon or a button, which performs an individually implementable function.

The input component can work together with other input components. In this way, e.g. the editor's input can be used for the script.

For information on the configuration of FS_BUTTON see FirstSpirit Online Documentation, "Template development" / "Forms" / "Input components (new)" / "FS_BUTTON" area.

If FS_BUTTON is used for application integration, i.e. if the AppCenter is used, a corresponding licence is required (see Chapter 6.5.2 page 137).

! *The input component FS_BUTTON is currently not supported in FirstSpirit WebClient. Appropriate support is planned for FirstSpirit Version 5.*

5.5 CMS_INCLUDE_OPTIONS: Restrict set of selectable elements

Using the data element CMS_INCLUDE_OPTIONS the value set which is to be displayed in the input components

- CMS_INPUT_CHECKBOX
- CMS_INPUT_COMBOBOX
- CMS_INPUT_RADIOBUTTON and
- CMS_INPUT_LIST

for selection can be configured dynamically. In this way, the input components can be, for example, filled with data records of an external table.

The tag `QUERY` can now be used to restrain the set of data records on the basis of a query which is defined for the respective database schema. The reference name of the desired query is to be indicated with the parameter `name`, e.g.

```
<CMS_INPUT_COMBOBOX name="IDENTIFIER" useLanguages="no">

    <CMS_INCLUDE_OPTIONS type="DATABASE">
      <LABELS>
        <LABEL lang="*">#item.COLUMN</LABEL>
        <LABEL lang="DE">#item.COLUMN_DE</LABEL>
      </LABELS>
      <TABLE>UNIQUE_NAME_TABLE_TEMPLATE</TABLE>
      <QUERY name="REFERENCE_NAME_QUERY"/>
    </CMS_INCLUDE_OPTIONS>

        <LANGINFOS>
            <LANGINFO lang="*" label="TEXT" description="TEXT"/>
            <LANGINFO lang="DE" label="TEXT" description="TEXT"/>
        </LANGINFOS>
</CMS_INPUT_COMBOBOX>
```

If there are parameters defined for the query specified by `QUERY` (tab "Parameters" of the query), you can access them by means of the tag `PARAM`. The name of the query parameter is indicated by means of the parameter `name`, the value of the query parameter within an opening and a closing `PARAM` tag, e.g.

```
<QUERY name="REFERENCE_NAME_QUERY">
    <PARAM name="PARAMETER_NAME">VALUE</PARAM>
</QUERY>
```

5.6 Functional enhancements of the Integrated preview

From 4.2R4, the "Integrated preview" introduced in JavaClient in the initial version of FirstSpirit 4.2 can not only be used for a preview of the currently edited content as to date, but also (if correctly configured (see Chapter 4.2.2.3 page 37)) other file formats (e.g. Microsoft Word or Excel files, see Chapter 4.2.2.1 page 34) and the FirstSpirit Online Documentation (see Chapter 4.2.2.4 page 39) can now also be displayed there, namely with the help of an application matching the format.

For template developers, particular reference is to be made to the display of the *FirstSpirit Online Documentation* (ODFS) in the Integrated preview. The current context for the display of a suitable page in ODFS is used within forms and output channels. For example, if you are in a form on a `CMS_INPUT_TEXT` tag, when the Online Help is opened (e.g. using F1), the "CMS_INPUT_TEXT" page (*Template development* / *Forms* / *Input components* area) is directly displayed in the ODFS, when called on a `$CMS_REF(...)$` tag in the output channel, the page "$CMS_REF(...)$" (*Template development* / *Template syntax* / *Instructions* area). If text is selected in the form or an output channel tab, and then the Online Help is opened, a search is automatically performed in the ODFS using the selected text and the results are displayed as usual. Furthermore, you can copy sections (for example sample code) from the online help to the form tab of templates simply by drag & drop.

For further information on this topic, see also Chapter 4.2.2 page 32.

5.7 Inline tables: CMS_INPUT_NUMBER in style templates

Since Version 4.1, so-called inline tables can be defined for JavaClient (DOM Editor with `table="yes"`). These are based, among other things, on style templates, via which input components can be created for maintaining layout attributes. Until now, the following input components only were supported:

- CMS_INPUT_TEXT / CMS_INPUT_TEXTAREA: Text field for entering a value, e.g. for the background colour of a cell.
- CMS_INPUT_COMBOBOX: Selection from a pre-defined set of values, e.g. for giving a background colour or the text alignment.
- CMS_INPUT_RADIOBUTTON: Selection from a pre-defined set of values, e.g. for giving a background colour or a text alignment.
- CMS_INPUT_TOGGLE: Choice between two possible values (e.g. on/off, right/left).

Since 4.2R4, the input component CMS_INPUT_NUMBER can also be used. For example, the value for the background colour of a cell can be given in this.

> ! *Within this framework, a so-called visibility of certain input components was introduced for style templates (see Chapter 5.8 page 101).*

5.8 Restrict visibility of input components

Within the framework of changes made for Chapter 5.7, a so-called visibility of certain input components for certain template types was introduced in 4.2R4. For example, the following tags only can now be used on the "Form" tab of style templates:

- CMS_COMMENT
- CMS_GROUP
- CMS_INPUT_COMBOBOX
- CMS_INPUT_NUMBER

- CMS_INPUT_RADIOBUTTON
- CMS_INPUT_TEXT
- CMS_INPUT_TEXTAREA
- CMS_LABEL
- CMS_MODULE

Style templates for which a different tag is used cannot be saved. The following information appears: "Bad GOM utilization! Tag 'CMS_INPUT_CONTENTAREALIST' not allowed in this kind of template! Editor 'xyz'".

The saving of invalid forms in table and link templates was also adjusted within this framework: all components except CMS_INPUT_SECTIONLIST can now be used for this.

In own input components and modules, the visibility must be adjusted accordingly using the `<scope>` or `<restrictions>` tag in the `fs-module.xml` file, as these will otherwise be rejected in links and style templates without change. The following values are available:

- *DATA*: The input component can be used in page and section templates.
- *CONTENT*: The input component can be used in table templates.
- *LINK*: The input component can be used in link templates.
- *STYLE*: The input component can be used in style templates.

Examples:

```
<restrictions>
     <kind>DATA</kind>
     <kind>CONTENT</kind>
     <kind>LINK</kind>
</restrictions>
```

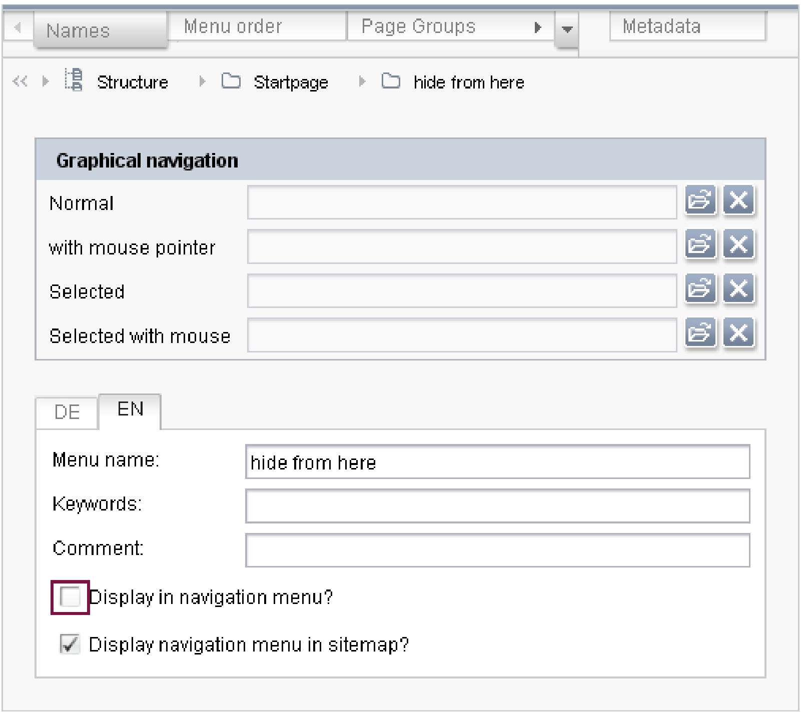
or

```
<scope data="yes" content="yes" link="yes" style="no"/>
```

An input component with this configuration could be used in page, section, table and link templates, but not in style templates. If nothing is given for an input component, it is allowed for the *DATA* and *CONTENT* scopes.

5.9 Navigation: `parentpath` hides menu levels if necessary

Whereas the key term `purepath` in combination with the `expansionVisibility` parameter of the navigation function (see *FirstSpirit Online Documentation*, *Template development / Template syntax / Functions / in the header / Navigation*) always displays the whole path to the current menu item regardless of the visibility, the key term `parentpath` introduced in 4.2R4 can be used to hide menu levels in the path. Namely, in this mode, menu levels (folders in the Site Store) are taken into account for which the "Display in navigation menu?" option is disabled on the "Names" tab:



Graphical navigation

Normal	
with mouse pointer	
Selected	
Selected with mouse	

Menu name:	hide from here
Keywords:	
Comment:	

☐ Display in navigation menu?

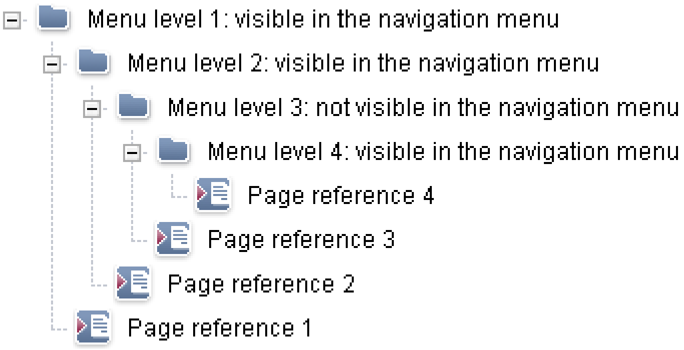
☑ Display navigation menu in sitemap?

Figure 5-5: Menu level hidden from the navigation menu

If a menu level is removed from the navigation menu in this way, and if `expansionVisibility="parentpath"` is set, starting from the root node (Site Store), the path is output only up to the last visible menu level. I.e., the first menu level starting from the root node with "Display in navigation menu?" option disabled and all menu levels below it are not displayed in the path.

Example:

In the case of a structure like



- Menu level 1: visible in the navigation menu
  - Menu level 2: visible in the navigation menu
    - Menu level 3: not visible in the navigation menu
      - Menu level 4: visible in the navigation menu
        - Page reference 4
      - Page reference 3
    - Page reference 2
  - Page reference 1

the menu levels: "Menu level 1" and "Menu level 2" would be visible.

5.10 Tagging

Since version 4.0 already FirstSpirit provides the option to supply revisions with so-called tags. These are special markings. From FirstSpirit Version 4.2R4 on, tags are created automatically at the following operations:

- when updating templates
- when importing packages (FirstSpirit PackagePool)
- when generating a project

However, these tags are only relevant for developers and they are no (yet) reflected in the FirstSpirit user interface. Tags can be created according to the following schema in a script by means of the API, for example:

```
Tag Project#createTag(String tag, String comment)
```

The tags of a project can be shown as follows:

```
List<Tag> Project#getTags(Tag maxTag, int maxCount)
```

The revision for a tag can be identified as follows:

```
de.espirit.firstspirit.storage.Tag
Revision getRevision()
```

By this means, a generation script can be created, for example, which identifies the revision for a tag and then generates this.

With

```
setCreateGenerateTag(...)
```

the tagging can be activated or deactivated for the generation, with

```
getCreateGenerateTag()
```

you can retrieve if tags are set.

It is planned for future FirstSpirit Versions to visualize tags in the user interface (e.g. in the version history) and maybe to realize editorial operations on tags.

5.11 Calling classes from modules

Own classes and methods in FirstSpirit modules can be called in (BeanShell) scripts. Additionally, classes can be executed directly in FirstSpirit Version 4.2. For this purpose, the class must implement the interface `de.espirit.firstspirit.access.script.Executable`

For this reason, the own class must provide implementations for the methods `execute(Map<String,Object> context)` and `execute(Map<String,Object> context, Writer out, Writer err)` (see FirstSpirit API documentation).

If the class can be found by the module classloader (i.e. as library) it can be executed in a script in FirstSpirit by calling:

```
-------------------------------------------------
#!executable-class
NAME_OF_CLASS
-------------------------------------------------
```

e. g.

```
-------------------------------------------------
#!executable-class
de.test.myClass
-------------------------------------------------
```

The class name must be given in the second line.

If `executable-class` is used in a task script, for example, transfer parameters can be deposited (in the Project properties "Schedule management" / activity "Execute Script" / button "Properties" / "Parameters", see also *FirstSpirit Manual for Administrators*).

These parameters can be accessed in the methods of `execute` (see above `Map<String, Object>`).

The advantage of this script is that in many cases it is not necessary to start external systems and the speed (e.g. of generation) increases compared to use of the Beanshell console.

This function is used for example in the FirstSpirit module "Portal SAP EP BP", namely in the "Portal_createLink" script (see also Chapter 7.1.2 page 156).

5.12 New and changed keyboard shortcuts

Lines on form and output channel tabs in the Template Store can now be removed using Ctrl + Shift + D; previously Ctrl + Y was used for this. On the other hand, Ctrl + Y (and with Ctrl + Shift + Z also) can now be used to restore ("Redo") changes undone using Ctrl + Z.

5.13 Changes in the API

Compared with FirstSpirit Version 4.2R2, no incompatible changes have been carried out to the public API in 4.2R4.

5.13.1 FirstSpirit Access-API

Several methods have been set to the "deprecated" status in FirstSpirit Version 4.2R4. Refer to the FirstSpirit Access API to find out which methods are involved and which methods are to be used as replacement. Methods with this status can be used indeed, but the use is disadvised because these methods will be omitted in FirstSpirit Version 5.0.

In addition, the following API enhancements have been made:

- Using the metadata of revisions, changes made at outgoing **references** for a revision are now available (see also *FirstSpirit Manual for Developers (Part 1: Basics)*, Chapter "Tracking Changes using Revision Metadata"). For this purpose, the methods of the interface `de.espirit.firstspirit.storage.ReferencesChangeDetail` can be used.
- If a **schedule** for which the "Not allowed (execute consecutively)" option is selected for "Parallel execution" in the Project properties under "Schedule Management" / "Configure" button, is repeatedly started, the list of schedules not yet executed can be returned using

```
de.espirit.firstspirit.access.schedule.ScheduleStorage#getQueuedEntries()
```

- The **evaluation of rights** when using the interface `de.espirit.firstspirit.access.AdminService` has been revised in 4.2R4: It is now explicitly checked, if the action which is carried out by the respective user is allowed to be executed because of his rights. Two types of action or user are differentiated here:

  - actions which require the rights of a project administrator
  - actions which require the rights of a server administrator

If the user does not have the necessary rights, a SecurityException will be returned. Methods which need rights of project or server administrators have got a corresponding notice about a SecurityException in the API. For example, the method `createLanguage` of the interface `de.espirit.firstspirit.access.AdminService` requires rights of a server administrator.

- In 4.2R4 the **creating operations** for IDProvider objects have been consolidated: The display name can be committed already when an object is created initially. Advantages:
  - The creating operation does not need a lock no more, i.e. the object will not be set into the editing mode. For assigning a display name it has been required so far to execute a lock-based change in addition after the creating operation, which does not need a lock. Using these new methods the display name can be assigned already with the creating operation without need for locking.
  - The additional revision and related data which have been created by the subsequent setting of the display name are eliminated.

  For example, a folder with a language-dependent display name (German and English) has been created in the Media Store so far as follows:

```
MediaStore ms;
MediaFolder newFolder = ms.createMediaFolder("myfolder", true);
try {
            newFolder.setLock(true, false);
            // set displayname for german
            LanguageInfo deLangInfo = newFolder.getLanguageInfo(germanLanguage);
            deLangInfo.setDisplayName("mein Ordner");

            // set displayname for english
            LanguageInfo enLangInfo = newFolder.getLanguageInfo(englishLanguage);
            enLangInfo.setDisplayName("my folder");

            newFolder.save("display names defined", false);
} finally {
            try {
          myLockFolder.setLock(false, false);
     } catch(Exception unlockFailed){
          // error handling
     }
}
```

From 4.2R4 it can be carried out in this way:

```
MediaStore ms;
Map<Language, String> lang2DisplayName = new HashMap<Language,
String>();
lang2DisplayName.put(germanLanguage, "mein Ordner");
lang2DisplayName.put(englishLanguage, "my folder");
MediaFolder newFolder = ms.createMediaFolder("myfolder",
lang2DisplayName, true);
```

Moreover, in FirstSpirit Version 4.2 Release 4 a first version of the **Application integration API** is available. It permits to realize individual Application integrations for FirstSpirit completely independently. This API has been released with Version 4.2R4, but it is still enhanced at the moment and will still change. Individual implementations which are carried out in 4.2R4 must be adapted to potential changes in the API most likely in a later version.

5.13.2 FirstSpirit Developer-API

For the first time, a Developer-API is delivered with FirstSpirit Version 4.2 Release 4. Unlike the Access-API (see Chapter 5.13.1 page 107), the Developer-API is subjected to less stability conditions: While the Access-API is stable within a major version line, the Developer-API is only stable within a **minor** version line, i.e. changes may be implemented in a minor version change.

The documentation for the FirstSpirit Developer-API can, e.g. be opened via the "Template development" / "FirstSpirit AP"I / "API Documentation" page in the *FirstSpirit Online Documentation*.

5.13.3 FirstSpirit Application-API

In addition, with FirstSpirit Version 4.2 Release 4, an initial version of the Application-API is provided, which enables fully independent individual application integrations to be implemented for FirstSpirit. This has emerged from practical experience of application integration of the core product and has therefore already been tried and tested in practice.

In conjunction with the add-on interface (FirstSpirit modules) also long established in the FirstSpirit product, product add-ons can be developed, which seamlessly integrate into the FirstSpirit application integration infrastructure.

The true potential of application integration therefore especially lies within the domain of FirstSpirit development partners for whom, with the help of the application integration API, a powerful tool has been made available, with which they can realise specialised integration solutions in their own particular areas of expertise.

At this point, however, it should be noted that application integration is a highly complex topic which, even after a development period of more than two years, is still not fully developed, as experience with it in turn opens up an increasing number of new applications.

Even at the level of the core product, development of FirstSpirit application integration has a long way to go before it is completed, and is still in full swing. Nevertheless, e-Spirit has made the decision to release an interface for application integration as early as possible, in order to allow partners to realise their own application integrations as quickly as possible.

However, this decision in favour of early API release was also accompanied by the need to water down the maxim of long-term safeguarding of interface stability explicitly defined when developing FirstSpirit: The interface made available with FirstSpirit Version 4.2 Release 4 will continue to be developed in following versions, and it will very probably become necessary for changes to be made to the module implementation. Each development partner who makes an implementation on the basis of the application integration API of FirstSpirit Version 4.2 Release 4 should be aware of this fact!

On the other hand, the opportunity is there to flexibly respond to and optimally support additional, previously unthought of use scenarios of application integration at the API level.

With FirstSpirit Version 5.0, a revised Version 2.0 of the API satisfying the long-term stability criteria will be available for application integration, and which will then be serviced and further developed according to the familiar strategies of maintaining stable APIs in FirstSpirit.

6 New/modified Functions for Administrators

6.1 Creating and using JDBC driver modules

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

Up to FirstSpirit Version 4.2R2 (inclusively) the required JDBC driver files were usually integrated into the FirstSpirit Server by filing them into the directory`.../shared/lib`.

From FirstSpirit Version 4.2R4 on, the integration of JDBC driver files as FirstSpirit module is advised. One advantage is for example, that different versions of one driver can be operated in parallel. Moreover, JDBC drivers can be exchanged while the FirstSpirit Server is running, without restart. For reasons of licencing these FirstSpirit modules are not delivered with the software but must be created by each customer himself. This Chapter explains how to proceed to switch from the use of JDBC drivers in the directory `.../shared/lib` to the integration as FirstSpirit module.

In this context, the use of databases

- on the Jetty web server (integrated in FirstSpirit),
- on a Tomcat web server (external),
- with Jetty and Tomcat web server

and

- the use of the Derby database in web applications (e.g. FirstSpirit module "Integration")

is referred to.

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

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

The FirstSpirit Manual about developing components (only German) provides further information about the development of modules in FirstSpirit in common, for further information about the configuration of databases, web servers and about installation and updating of FirstSpirit modules see "FirstSpirit Manual for administrators".

6.1.1 Creating a JDBC driver module

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

- file(s) of the JDBC driver, e.g. JAR file, licences etc. (see Chapter 6.1.1.1 page 112)
- file `module.xml` (see Chapter 6.1.1.2 page 113)

optionally:

- file `web.xml` (see Chapter 6.1.1.2 page 113)

These files must be integrated by means of an FSM file (ZIP archive) to a FirstSpirit module (see Chapter 6.1.1.3 page 117).

6.1.1.1 Files of the driver

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

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

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

must be added in the database configuration.

6.1.1.2 module.xml and web.xml

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

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

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

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

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

```
    </components>

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

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

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

`<name>`: This tag must be used for assigning a unique name for the components. Only the following characters are allowed:
capital and lower case letters (A-Z, a-z), figures (0-9) and the special characters ; , _ \ -
The name which is assigned for the server component will be also used in the overview of modules of the FirstSpirit Server (see Figure 6-3) and must be indicated in the database layer configuration, too (see Chapter 6.1.2.2 page 119). The name which is assigned to the web application component will be also used in the Project properties, area "Web components" (see Figure 6-7). For a clearer overview *_Server* and *_WebApp* have been attached in the example to the names of the server and the web application component. .

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

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

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

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

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

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

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

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

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

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

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

for PostgreSQL or

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

for Oracle.

`<DRIVER>`: Contains the complete class name of the used JDBC driver, e.g. `org.postgresql.Driver` for PostgreSQL.

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

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

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

If the integrated **Derby database** is to be used in the web applications of a Tomcat web server, you need also a `module.xml` file. An exemplary `module.xml` could look like this:

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

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

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

     </components>

   <configuration>
      <layerclass>de.espirit.or.impl.derby.DerbyLayer</layerclass>
      <DRIVER>org.apache.derby.jdbc.ClientDriver</DRIVER>
   </configuration>
</module>
```

The part for the server component is only then required if both the internal Jetty and an external Tomcat web server are used at the same time.

The file `web.xml` is required as well. Example:

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

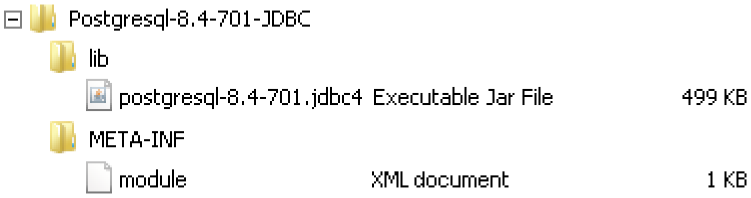
If the part for the server component is required, i.e. if internal Jetty and external Tomcat web server are used at the same time, the database layer must be adjusted (see Chapter 6.1.2.4.1 page 123).

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

An explication for the most tags used in these examples can be looked up also in the *FirstSpirit Manual for Developers (Components)* (German only).

6.1.1.3 Directory structure of a JDBC driver module

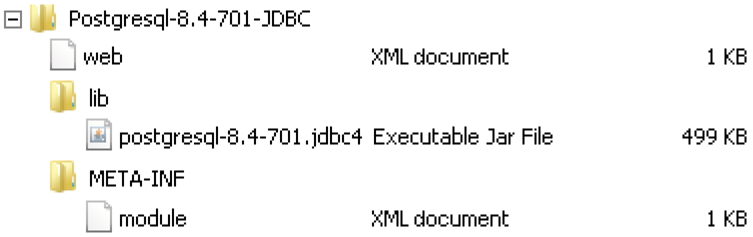
If the driver is only used for the FirstSpirit Server ("server component"), the files must be deposited in the following directory structure (cf. Chapters 6.1.1.1 and 6.1.1.2):



```
⊟ Postgresql-8.4-701-JDBC
    lib
        postgresql-8.4-701.jdbc4  Executable Jar File        499 KB
    META-INF
        module                    XML document                 1 KB
```

Figure 6-1: Directory structure server component

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



```
⊟ Postgresql-8.4-701-JDBC
    web                          XML document              1 KB
    lib
       postgresql-8.4-701.jdbc4  Executable Jar File     499 KB
    META-INF
       module                    XML document              1 KB
```

Figure 6-2: Directory structure server and web application component

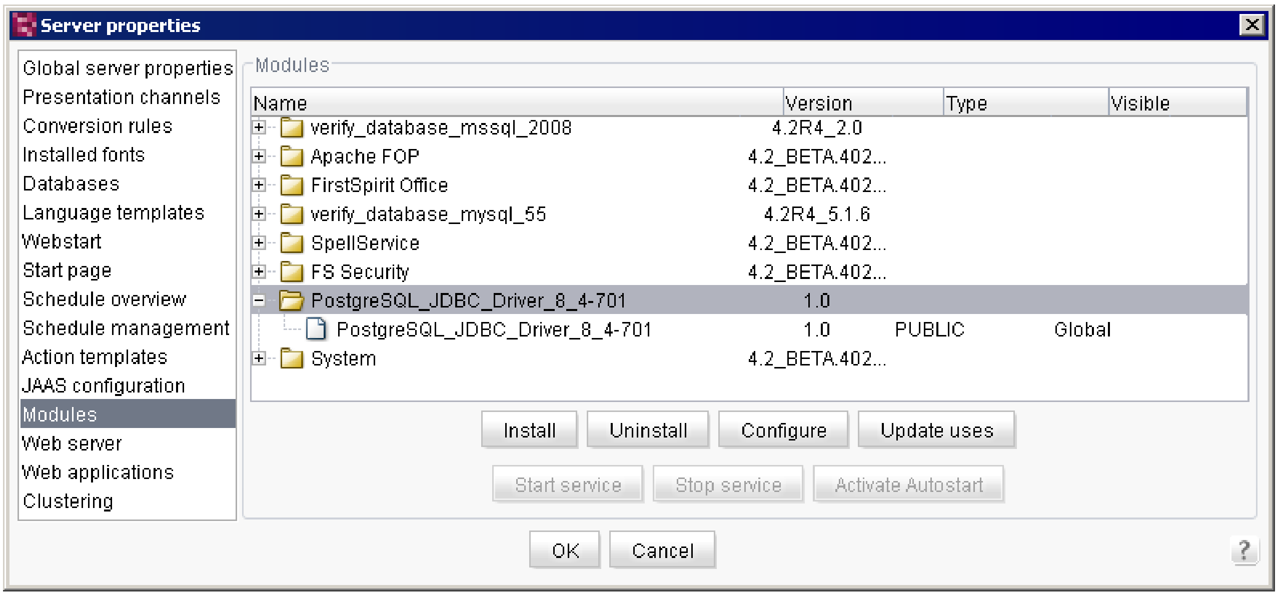
To get a valid FirstSpirit module, a ZIP file must be created from the content of the superordinate folder ("Postgresql-8.4-701-JDBC"). The superordinate folder must not be included in the ZIP file. This ZIP file must then be renamed into `*.fsm`. If, in the example of Figure 6-2, the folder name was taken as file name, the module file should have the name `Postgresql-8.4-701-JDBC.fsm`.

6.1.2 Installation and configuration of the JDBC driver module

6.1.2.1 Installation of the JDBC driver module

If the JDBC driver module has been created successfully as described in Chapter 6.1.1 from page 112, it must be installed on the FirstSpirit Server. This will be carried out by means of the application for the Server and Project configuration.

For this purpose, the button "Install" must be clicked in the Server properties in the area "Modules". The driver module file can be selected from the locale file directory in the following dialog and uploaded to the server. The successfully installed file will then be displayed in the overview as module with its name (here: `PostgreSQL_JDBC_Driver_8_4-701`, see Chapter 6.1.1.2 page 113, tag `<name>`), version (tag `<version>`) and the included component (tag `<components>`):



Server properties

- Global server properties
- Presentation channels
- Conversion rules
- Installed fonts
- Databases
- Language templates
- Webstart
- Start page
- Schedule overview
- Schedule management
- Action templates
- JAAS configuration
- Modules
- Web server
- Web applications
- Clustering

Modules

Name	Version	Type	Visible
verify_database_mssql_2008	4.2R4_2.0		
Apache FOP	4.2_BETA.402...		
FirstSpirit Office	4.2_BETA.402...		
verify_database_mysql_55	4.2R4_5.1.6		
SpellService	4.2_BETA.402...		
FS Security	4.2_BETA.402...		
PostgreSQL_JDBC_Driver_8_4-701	1.0		
PostgreSQL_JDBC_Driver_8_4-701	1.0	PUBLIC	Global
System	4.2_BETA.402...		

Figure 6-3: Server properties – JDBC driver as module

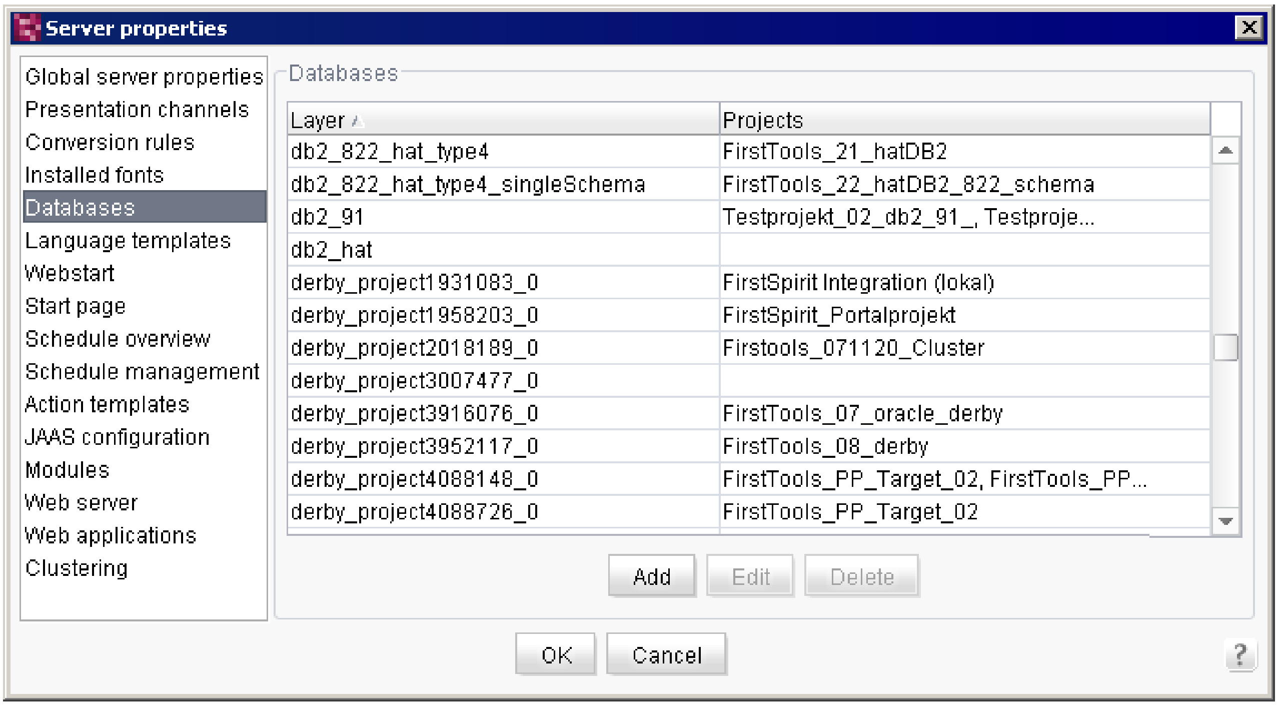
If the `module.xml` file contains the definition for a web application, this will be displayed here as well.

Here, no further configuration is required.

6.1.2.2 Configuration of the database layer

Subsequently, the parameter `module` must be used to refer to this driver module in the configuration of the layer of the database for which the driver module was created.

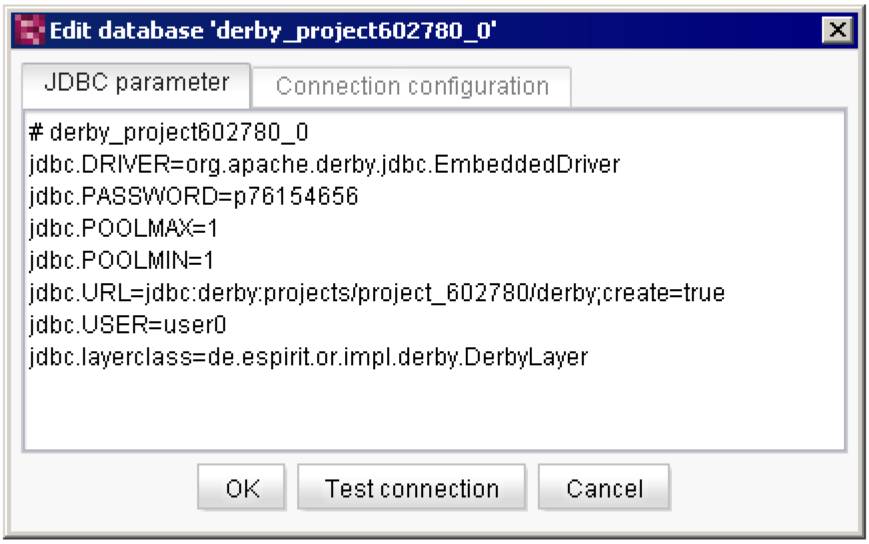
Select the respective database in the Server properties in the area "Databases":



Layer	Projects
db2_822_hat_type4	FirstTools_21_hatDB2
db2_822_hat_type4_singleSchema	FirstTools_22_hatDB2_822_schema
db2_91	Testprojekt_02_db2_91_, Testproje...
db2_hat	
derby_project1931083_0	FirstSpirit Integration (lokal)
derby_project1958203_0	FirstSpirit_Portalprojekt
derby_project2018189_0	Firstools_071120_Cluster
derby_project3007477_0	
derby_project3916076_0	FirstTools_07_oracle_derby
derby_project3952117_0	FirstTools_08_derby
derby_project408148_0	FirstTools_PP_Target_02, FirstTools_PP...
derby_project4088726_0	FirstTools_PP_Target_02

Figure 6-4: Server properties – Databases

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

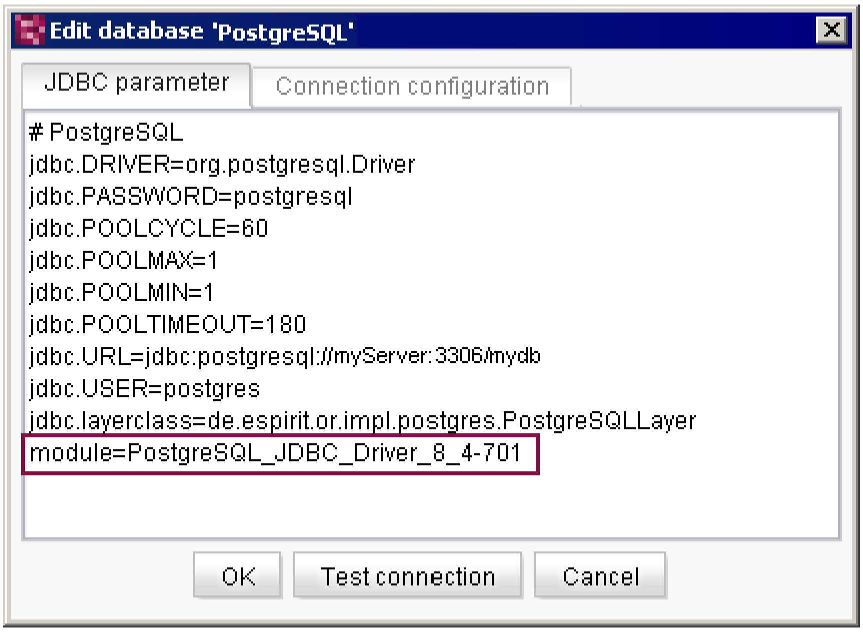


```
# derby_project602780_0
jdbc.DRIVER=org.apache.derby.jdbc.EmbeddedDriver
jdbc.PASSWORD=p76154656
jdbc.POOLMAX=1
jdbc.POOLMIN=1
jdbc.URL=jdbc:derby:projects/project_602780/derby;create=true
jdbc.USER=user0
jdbc.layerclass=de.espirit.or.impl.derby.DerbyLayer
```

Figure 6-5: Database configuration

Add the parameter `module` with the name which has been defined via `<name>` in the `module.xml` file, in our example:

```
module=PostgreSQL_JDBC_Driver_8_4-701
```



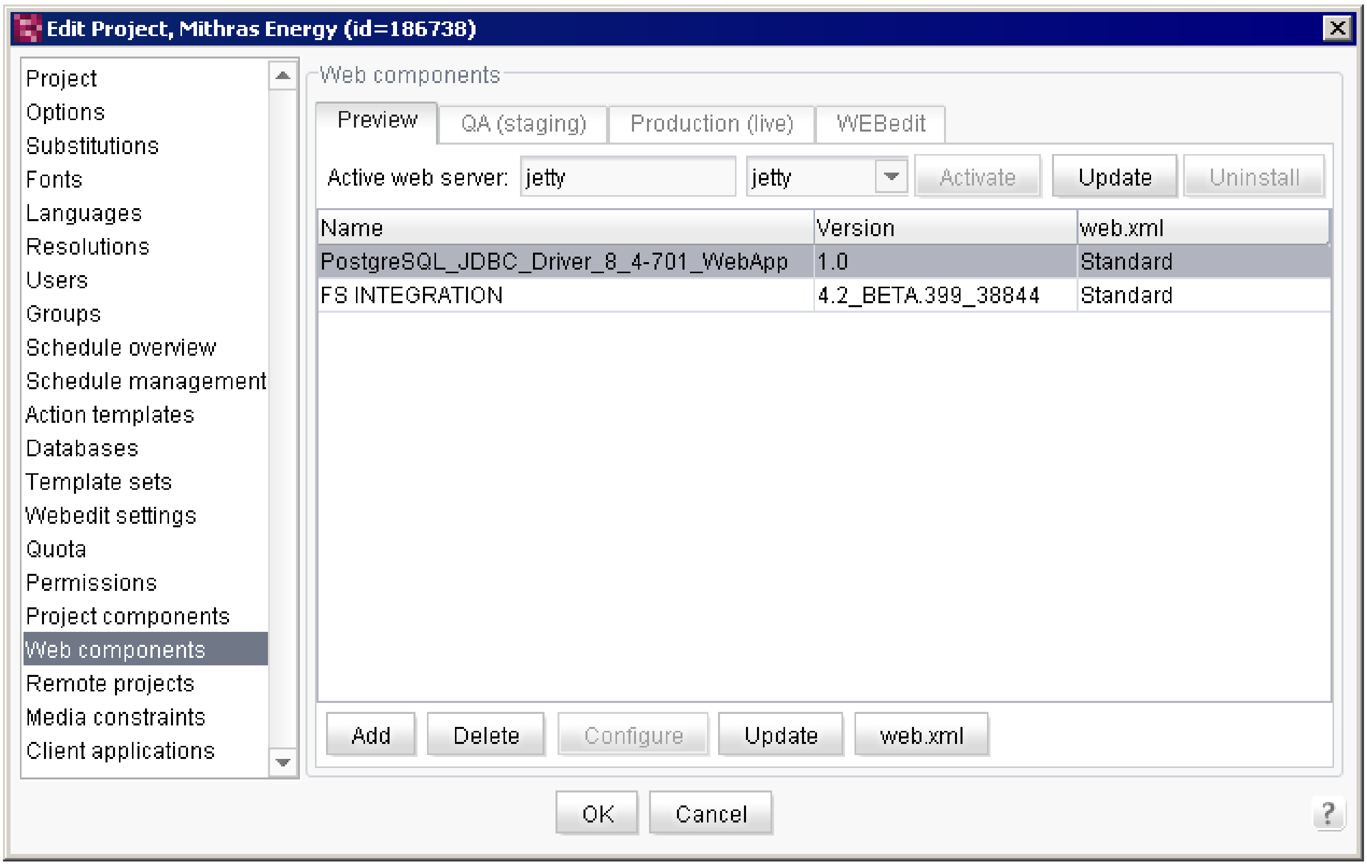
```
# PostgreSQL
jdbc.DRIVER=org.postgresql.Driver
jdbc.PASSWORD=postgresql
jdbc.POOLCYCLE=60
jdbc.POOLMAX=1
jdbc.POOLMIN=1
jdbc.POOLTIMEOUT=180
jdbc.URL=jdbc:postgresql://myServer:3306/mydb
jdbc.USER=postgres
jdbc.layerclass=de.espirit.or.impl.postgres.PostgreSQLLayer
module=PostgreSQL_JDBC_Driver_8_4-701
```

Figure 6-6: Database configuration with parameter `module`

This modification can be saved after a successful test of the connection with a click on the button "OK".

6.1.2.3 Usage in web applications

If the database is to be used in a web application, the module must be added to the desired web component. This is effectuated in the Project properties in the area "Web components". Click on "Add". A list will open from which you can select the module components which are available on the server. Select the component of the JDBC driver module. The name results from the value given by means of `<name>` within the definition of `<web-app>` (see Chapter 6.1.1.2 page 113):



Edit Project, Mithras Energy (id=186738)

Web components

Active web server: jetty | jetty

Name	Version	web.xml
PostgreSQL_JDBC_Driver_8_4-701_WebApp	1.0	Standard
FS INTEGRATION	4.2_BETA.399_38844	Standard

Figure 6-7: Project properties – JDBC driver as web component

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

6.1.2.4 Usage of the Derby database in web applications

If a Derby database is used in web applications (e.g. in the FirstSpirit module "Integration"), the JDBC driver module must be added to the web application as well, as described in Chapter 6.1.2.3 page 121.

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

6.1.2.4.1 Example: Module "FirstSpirit Integration"

When using the module "FirstSpirit Integration" with a Tomcat web server, the connection configuration must be adjusted for each schema. In this case, the Derby database can be only accessed by means of the TCP port. For this purpose, the parameter `internalDB.port` must be indicated in the configuration file `fs-server.conf`.

In addition, the following parameters must be adjusted in the Configuration of the database layer (see Chapter 6.1.2.2 page 119) for each schema:

`jdbc.URL`: This parameter must point to the TCP port of the Derby database instead of a locale directory. For this purpose, host and port must be added in the existing URL and `create` must be deleted, e.g.

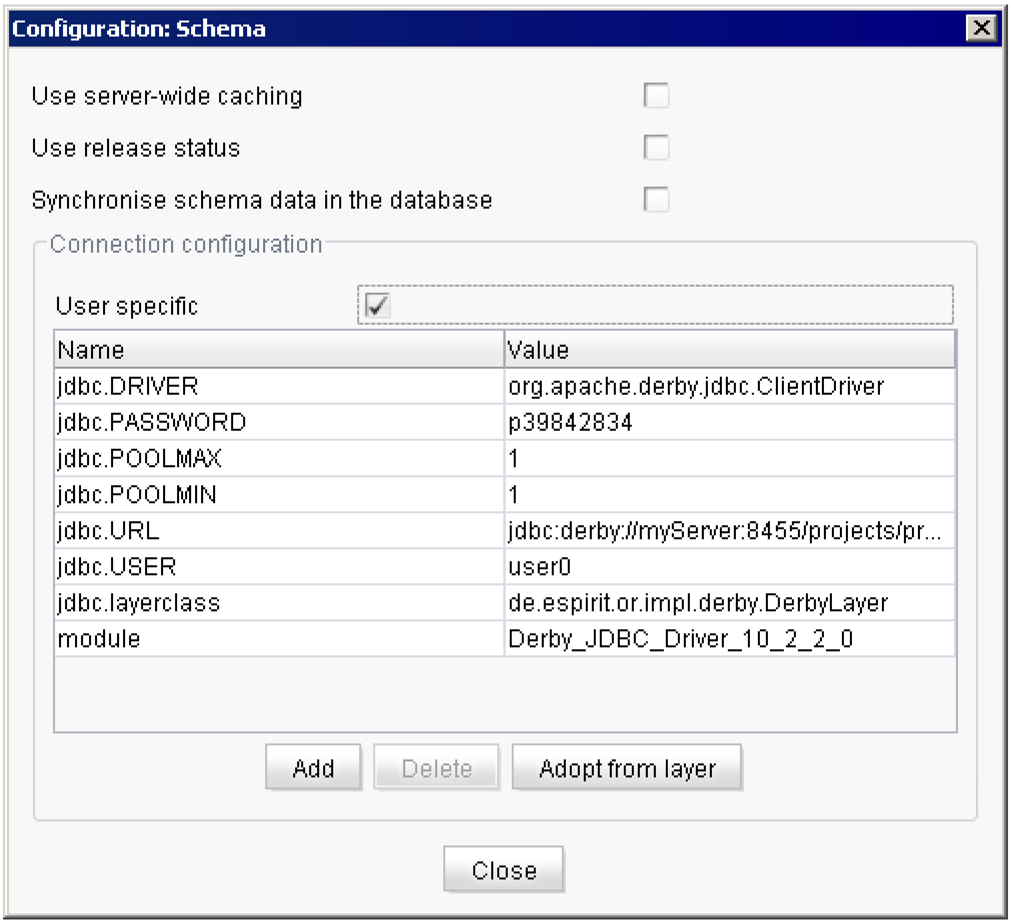
```
jdbc:derby:projects/project_29703/derby;create=true
```

will become

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

`jdbc.DRIVER`: Change this parameter to `org.apache.derby.jdbc.ClientDriver` if you use a Tomcat web server. When using a Jetty web server no adjustment is necessary.

If you use the FirstSpirit module "Integration" the configuration of the JDBC driver module must be updated manually after these modifications, if the option "User specific" is activated in the database connection:



Configuration: Schema

Use server-wide caching □

Use release status □

Synchronise schema data in the database □

Connection configuration

User specific ☑

Name	Value
jdbc.DRIVER	org.apache.derby.jdbc.ClientDriver
jdbc.PASSWORD	p39842834
jdbc.POOLMAX	1
jdbc.POOLMIN	1
jdbc.URL	jdbc:derby://myServer:8455/projects/pr...
jdbc.USER	user0
jdbc.layerclass	de.espirit.or.impl.derby.DerbyLayer
module	Derby_JDBC_Driver_10_2_2_0

Figure 6-8: FirstSpirit Integration – Database configuration

If this option is not activated, the values which are stored currently in the JDBC driver module and in the configuration of the database layer will be assumed.

If the internal Jetty web server is used in parallel to the external Tomcat web server, the database layer must be adjusted as well (see Chapter 6.1.2.2 page 119).

6.1.2.4.2 Individual implementations of modules

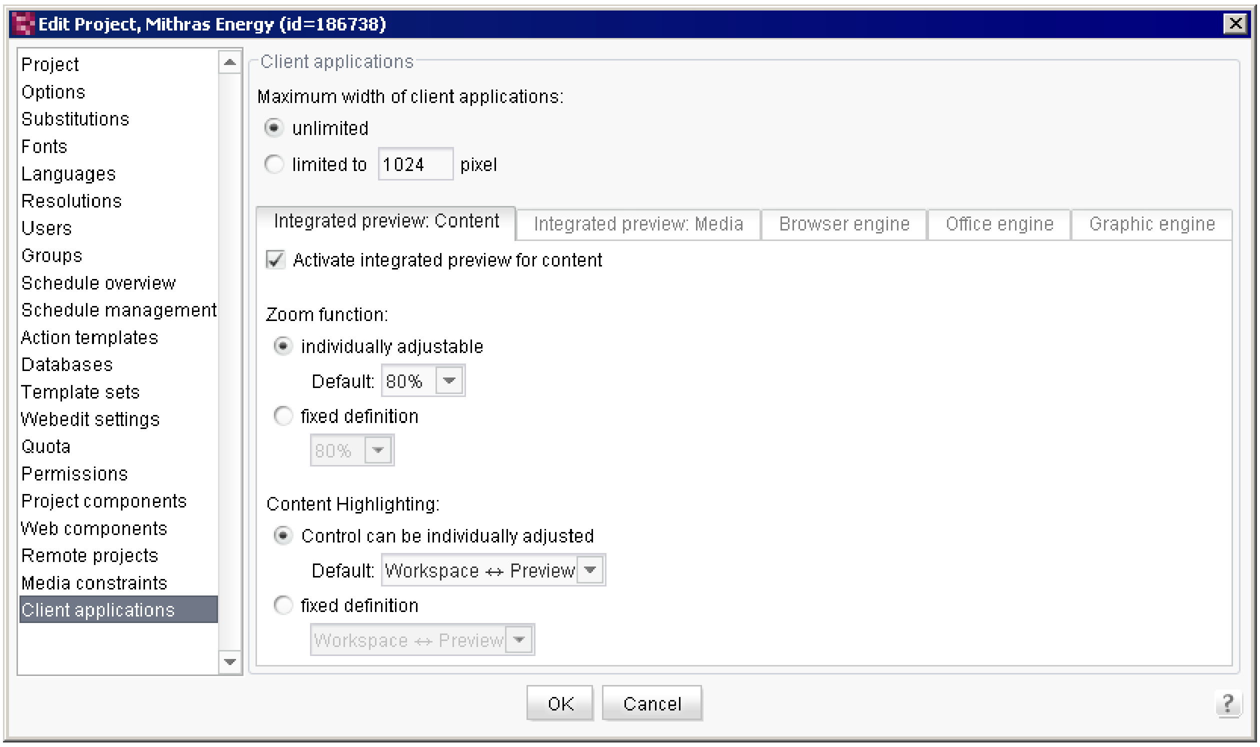
You must carry out these adjustments, which are carried out for the example "FirstSpirit Integration" (see Chapter 6.1.2.4.1 page 123), in each module which has been developed individually and which works together with the Derby database.

6.2 Functional enhancements of the Integrated preview

From 4.2R4, the "Integrated preview" introduced in JavaClient in the initial version of FirstSpirit 4.2 can not only be used for a preview of the currently edited content as to date, but also (if correctly configured (see Chapter 4.2.2.3 page 37)) other file formats (e.g. Microsoft Word or Excel files, see Chapter 4.2.2.1 page 34) and the FirstSpirit Online Documentation (see Chapter 4.2.2.4 page 39) can now also be displayed there, namely with the help of an application matching the format. The relative application must be installed on the workstation computer of the respective user to this end.

In order to distinguish between the Integrated preview in the Media Store from the Integrated preview for HTML content familiar to date, the HTML preview used in the right-hand column of JavaClient to date is called "Integrated preview for content", the new preview as the "Integrated preview for media".

Project-wide display options can be preset in the Project properties, namely in the "Client applications" area. The settings made here affect the "View" menu in JavaClient (see Chapter 4.2.2.3 page 37). The project concerned must be restarted in order for the changes to be adopted.



Client applications

Maximum width of client applications:

● unlimited

○ limited to 1024 pixel

☑ Activate integrated preview for content

Zoom function:

● individually adjustable

Default: 80%

○ fixed definition

80%

Content Highlighting:

● Control can be individually adjusted

Default: Workspace ↔ Preview

○ fixed definition

Workspace ↔ Preview

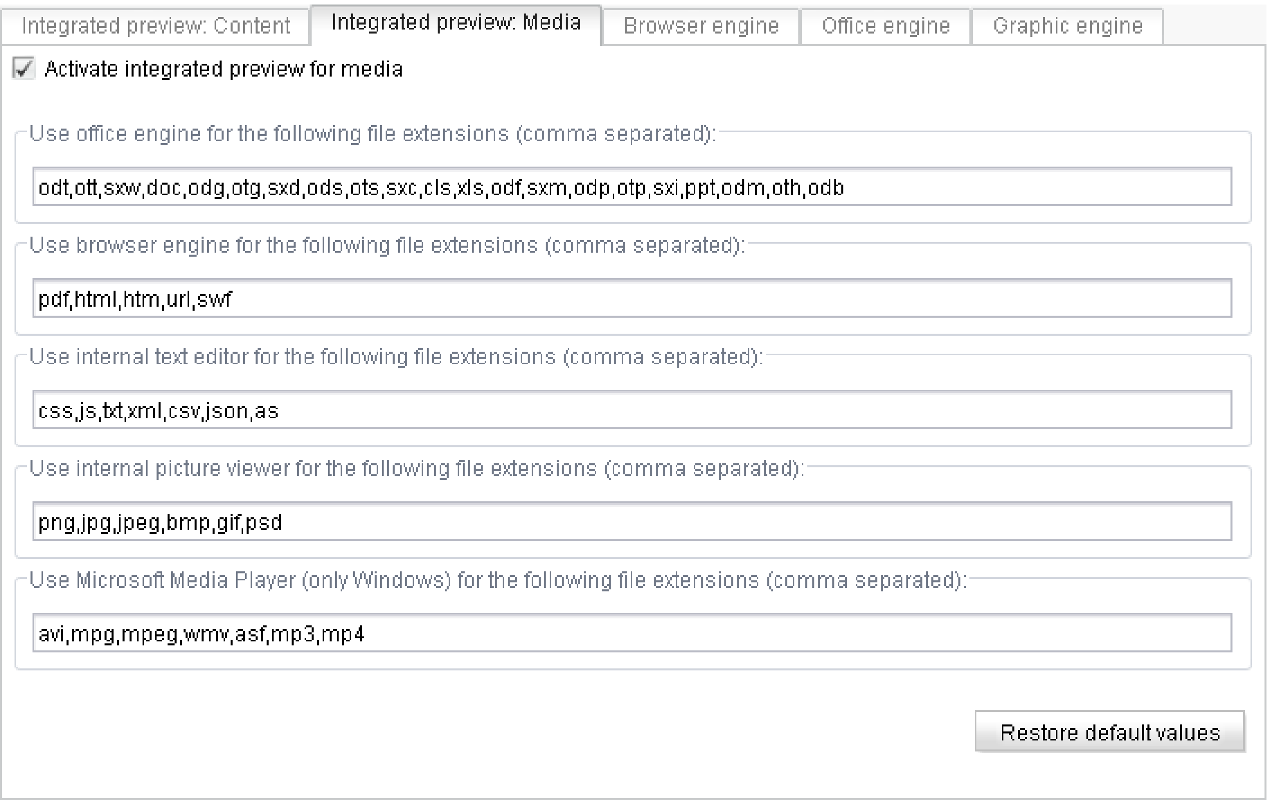
Figure 6-9: Project properties – Client applications

With 4.2R4 the JxBrowser which is used for the Integrated preview has been updated to the version 2.5 (with Gecko Engine 1.9.2 / Firefox 3.6). In some cases this update can lead to errors (e.g. `java.lang.UnsatisfiedLinkError`), and the Integrated preview will not be displayed. This problem can be solved by installing the Microsoft Visual C++ 2008 SP1 Redistributable Package (x86).

The following tabs are new in Version 4.2R4:

6.2.1 Integrated preview: Media

Here it is defined whether the Integrated preview for media is to be used in the respective project and which file formats are to be displayed by means of which application:



☑ Activate integrated preview for media

Use office engine for the following file extensions (comma separated):
odt,ott,sxw,doc,odg,otg,sxd,ods,ots,sxc,cls,xls,odf,sxm,odp,otp,sxi,ppt,odm,oth,odb

Use browser engine for the following file extensions (comma separated):
pdf,html,htm,url,swf

Use internal text editor for the following file extensions (comma separated):
css,js,txt,xml,csv,json,as

Use internal picture viewer for the following file extensions (comma separated):
png,jpg,jpeg,bmp,gif,psd

Use Microsoft Media Player (only Windows) for the following file extensions (comma separated):
avi,mpg,mpeg,wmv,asf,mp3,mp4

Figure 6-10: Project properties – Presettings Integrated preview/Media

Activate integrated preview for media: This option is related to the "Activate office engine" option on the "Office engine" tab (see Chapter 6.2.3 page 129): it is only possible to enable or disable both options simultaneously. If this option is disabled, no further changes can be made on this tab. The "use for media" option is greyed out for this project in the "View" / "Integrated preview" menu in JavaClient and cannot be selected. In new projects, which are created with a FirstSpirit Server

Version higher than 4.2.400, this option is activated by default. Thus, the following fields can be used to define which file format is to be displayed by which application in the Integrated preview. In projects which were created with a FirstSpirit Version lower than 4.2.400 this option is deactivated by default.

Use office engine for the following file extensions (comma separated): File extensions which are to be displayed via the Office engine can be given, comma separated, in this field. As a default, file formats which can be created and edited with Microsoft Office or OpenOffice are specified here.

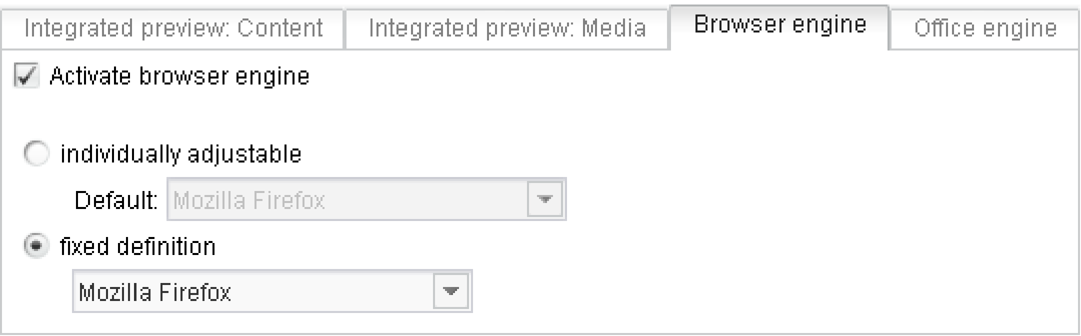
Use browser engine for the following file extensions (comma separated): File extensions which are to be displayed via the Browser engine can be given, comma separated, in this field. As a default, file formats which can be displayed by the Microsoft Internet Explorer and Mozilla Firefox web browsers or by plug-ins of these browsers are specified here.

Use internal text editor for the following file extensions (comma separated): File extensions which are to be displayed via the text editor integrated in FirstSpirit can be given, comma separated, in this field. As a default, file formats which can be created and edited with text editors are specified here.

Use Microsoft Media Player (only Windows) for the following file extensions (comma separated): File extensions which are to be played back via the Windows Media Player can be given, comma separated, in this field. As a default, audio and video file formats are specified here. The Windows Media Player can only be used in conjunction with Microsoft Windows; this field can be greyed out for other operating systems.

Restore default values: Click this button to restore the default setting.

6.2.2 Browser engine



☑ Activate browser engine

○ individually adjustable

Default: Mozilla Firefox

● fixed definition

Mozilla Firefox

Figure 6-11: Project properties – Presettings Integrated preview/Browser

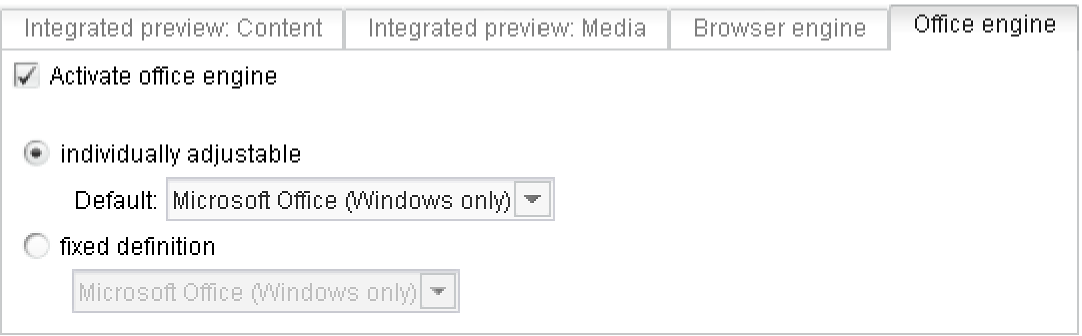
Activate browser engine: This option is related to the "Integrated preview for Content" option on the "Integrated preview: Content" tab: it is only possible to enable or disable both options simultaneously. No changes can be made on this tab if the "Activate browser engine" option is disabled. The file formats defined on the "Integrated preview: Media" tab in the "Browser engine" field (see Chapter 6.2.1 page 126), are then not displayed in the browser engine integrated in FirstSpirit, but are displayed in an external application instead. The option is enabled as a default, so that the following radio and comboboxes can be used to define which browser engine is to be used for the file formats defined in the "Browser engine" field:

individually adjustable: If this option is enabled, users can select the browser engine themselves in the relevant project. As a default, this option is enabled. The "Default:" combobox can be used to specify which browser engine is to be preselected in the project. As a default, this is Mozilla Firefox.

fixed definition: If this option is enabled, a browser engine can be specified for the project. The user can then not make any selection in this project in JavaClient, in the "Browser engine" sub-menu in the "View" menu.

For system requirements and restrictions concerning the browser engine see Chapter 3.1.3 page 20.

6.2.3 Office engine



☑ Activate office engine

● individually adjustable

Default: Microsoft Office (Windows only)

○ fixed definition

Microsoft Office (Windows only)

Figure 6-12: Project properties – Presettings Integrated preview/Office

A valid licence is required for using this function: The parameter `license.OFFICE_INTEGRATION` in the licence file `fs-license.conf` must be set to the value 1 (see also Chapter 6.5 page 136). Otherwise, the tab "Office engine" is greyed out, no settings can be made and Office documents can not be displayed in the Integrated preview of the JavaClient.

Activate office engine: This option is related to the "Enable Integrated preview for Media" option on the "Integrated preview: Media" tab (see Chapter 6.2.1 page 126): it is only possible to enable or disable both options simultaneously. If this option is disabled, no changes can be made on this tab. The file formats defined on the "Integrated preview: Media" tab in the "Office engine" field (see Chapter 6.2.1 page 126), are then not displayed in the Integrated preview for media, but are displayed in an external application instead. The option is enabled as a default, so that the following radio and comboboxes can be used to define which Office application installed on the workstation computer is to be used for the file formats defined in the "Office engine" field:

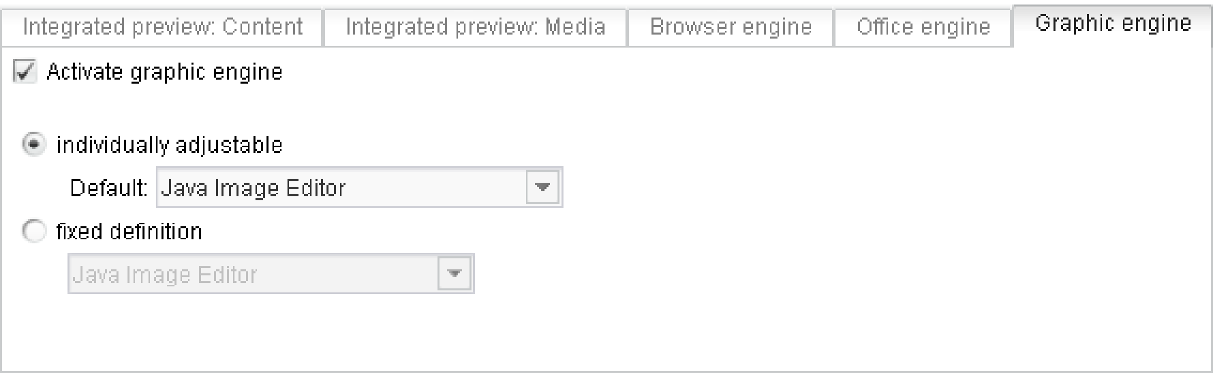
individually adjustable: If this option is enabled, users can select the Office engine themselves in the relevant project. As a default, this option is enabled. The "Default:" combobox can be used to specify which Office engine is to be preselected in the project. As a default, this is Microsoft Office. Microsoft Office cannot be used for non-Windows operating systems, therefore, "OpenOffice" (currently BETA test stage only!) should be selected here.

fixed definition: If this option is enabled, an Office engine can be specified for the project. The user can then not make any selection in this project in JavaClient, in the

"Office engine" sub-menu in the "View" menu.

Notes about the use of Microsoft Office see Chapter 3.1 page 17, about the use of OpenOffice and Google Docs see Chapter 3.3 page 22.

6.2.4 Graphic engine



Integrated preview: Content | Integrated preview: Media | Browser engine | Office engine | Graphic engine

☑ Activate graphic engine

● individually adjustable

Default: Java Image Editor

○ fixed definition

Java Image Editor

Figure 6-13: Project properties – Presettings Integrated preview/Graphic

Activate Graphic Engine: This option is related to the "Activate integrated preview for media" option on the "Integrated preview: media" tab (see Chapter 6.2.1 page 126): if the "Activate Graphic Engine" option is activated, the "Activate integrated preview for media" options is automatically activated too.
If the "Activate Graphic Engine" option is deactivated, changes cannot be made on this tab. The "Graphic Engine" entry in the "View" menu in JavaClient is greyed out and deactivated. Images can continue to be edited using the functions familiar to date (icon).
If the option is activated, the following radio buttons and comboboxes can be used to specify which applications are to be used:

individually adjustable: If this option is activated, users can select the graphic engine themselves in the relevant project. The "Default:" combobox can be used to specify which graphic engine is to be preselected in the project.

Fixed definition: If this option is activated, a graphic engine can specified for the project. Users can then not make a selection in the "Graphic Engine" submenu in the "View" menu in JavaClient in this project.

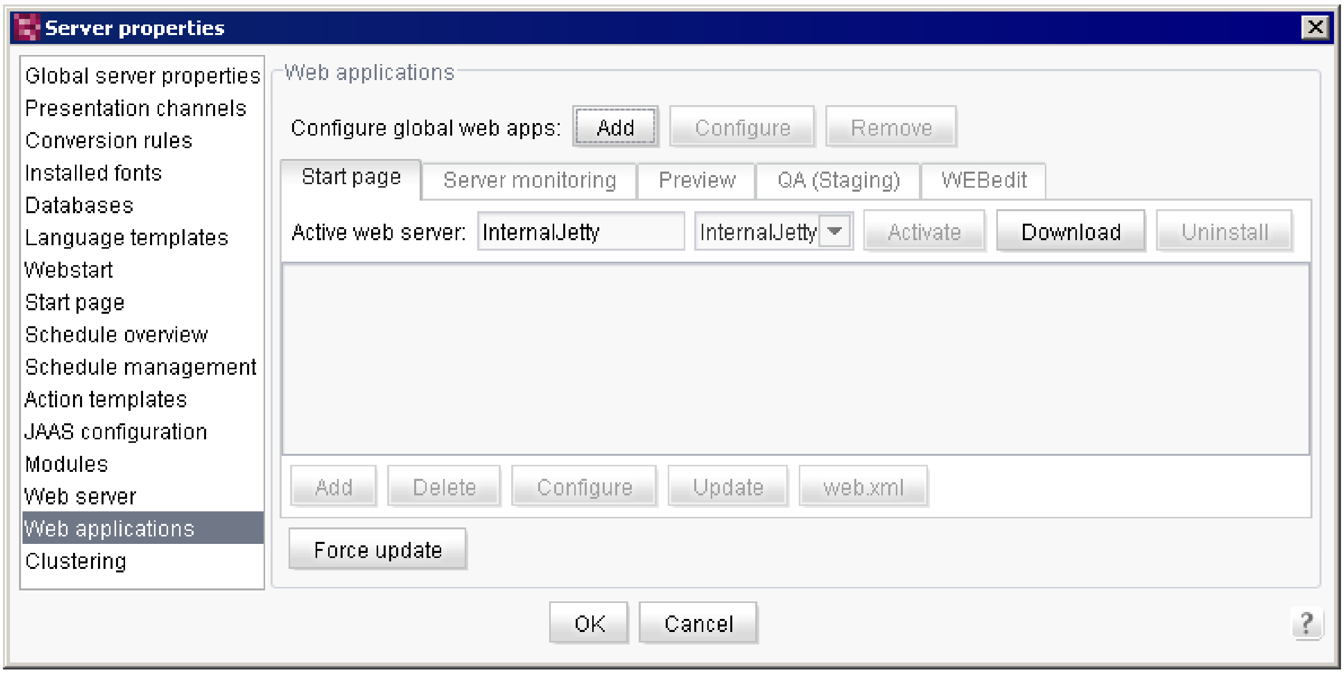
For further information on the image editing functions see Chapter 4.2.3 page 40. For details of system requirements and limitations with regard to the graphic engine, see Chapter 3.2 page 21 (Java Image Editor) and Chapter 3.3 page 22 (Picnik and

Pixlr).

6.3 Global web applications

From Version 4.2R4 and higher, the server properties can be used to define and configure the customer's own web applications, which are then available in all projects on the server ("global"). In this way, e.g. the customer's own web applications can be installed for the FirstSpirit AppCenter (see Chapter 3 page 16).

To do this, the server properties, area "Web applications", are opened:

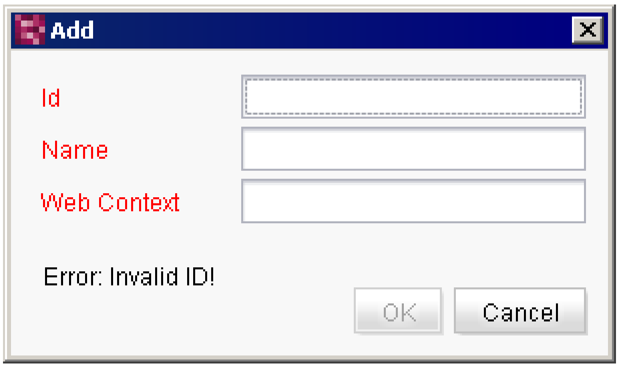


Active web server: InternalJetty InternalJetty

Figure 6-14: Server properties – Web applications

6.3.1 Installing global web applications

The "Add" button can be used to install a new web application on the server:



Error: Invalid ID!

Figure 6-15: Add new web application

Id: A unique identifier for the web application must be entered here. It can be accessed via the API (see Chapter 6.3.4 page 133). In addition, a subdirectory with this identifier is created on the server for the web applications. Only upper case, lower case letters, numbers and underscores may be used. This identifier can no longer be changed after it has been saved.

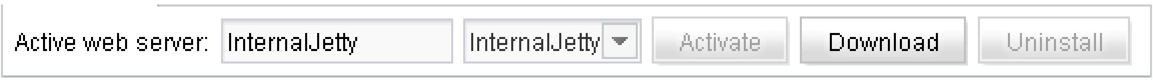
Name: A name must be given here, which is used as the display name. This can be changed later if and when necessary.

Web Context: A so-called context name must be entered here, which forms part of the URL to the web application. It may not be the same as the name of the existing standard FirstSpirit web applications (i.e. *fs4root*, *fs4preview*, *fs4staging*, *fs4webedit*, *fs4webmon*).

6.3.2 Configuration

For each web application

- a web server can be set:



Active web server: InternalJetty

- web components available on the server can be added, removed, configured and updated:

- the file `web.xml` can be manually edited:

The web components can be grouped together to form a web application and be installed on the web server or downloaded as a WAR file.

The function is analogous to that of web components in the project properties. See also *FirstSpirit Manual for Administrators*, "Web components" chapter.

6.3.3 Updating

The "Update uses" button is available in the server properties, "Modules" area, for updating global web applications. When the button is clicked, a dialog box opens for updating the web applications for all projects which have used this application to date. The updating therefore no longer has to be carried out individually for each product via the project properties, but instead can be controlled centrally via the server properties.

6.3.4 Access via the API

With the FirstSpirit Developer-API, the web application Id (see Figure 6-15) can be used to determine the URL to a global web application via the interface `de.espirit.firstspirit.agency.LegacyModuleAgent`.

6.4 Performance and security

In Version 4.2R4, changes were made with which it was able to increase the efficiency and speed of FirstSpirit, especially in the case of generations: The speed of generations could be increased for big projects by circa one third. Equally, more efficient data formats have been introduced for references (reference graph) and workflows (see also Chapter 2.2.1 page 13). References are also compressed on saving, if they exceed a certain threshold value.

In order to better utilise the existing hardware, the default values for the size of the thread pool and thread queue were also adjusted in the configuration file `fs-server.conf` for an improved performance. Therefore, following installation or a change of hardware, the FirstSpirit Server is now automatically scaled to the number of existing computing cores. The number of available processors is determined via

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

The default values in the **Thread Pool** area of the `fs-server.conf` file now look like this:

```
###########################
# Thread Pool
###########################
# minimum number of concurrent threads, if left empty the value is
# set to #cores (= number of cores as delivered by
# "java.lang.Runtime.availableProcessors()")
ThreadPool.minSize=
# maximum number of concurrent threads, if left empty the value is
# set to (#cores * 8)
ThreadPool.maxSize=
```

If no explicit values are given for the following parameters

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

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

The **Thread Queues** area of the `fs-server.conf` file now looks like this:

```
############################
# Thread Queues:
# - LOW: Queue for resource-intensive tasks.
# - DEFAULT: Default queue for default tasks.
# - BOUNDED: Bounded queue with rejection strategy.
#   (queueCapacity: -1 = unbounded, 0 = no queueing allowed)
# Attributes:
# - maxRunning   maxmimum numbers of running tasks.
# - queueCapacity queue capacity (-1 = unbounded, 0 = no queueing
# allowed).
# - rejection    rejection strategy for incoming tasks if queue
# capacity is exceeded (REJECT, BLOCK, EXECUTE).
#                REJECT reject task if queue capacity is exceeded
# (only allowed in BOUNDED queue!).
#                BLOCK  wait until queue has free capacities.
#                EXECUTE execute queue's oldest task inline.
############################
ThreadQueue.LOW.maxRunning=2
ThreadQueue.LOW.queueCapacity=128
ThreadQueue.LOW.rejection=BLOCK

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

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

If no explicit values are given for the following parameters

- *`ThreadQueue.DEFAULT.maxRunning`* (maximum number of tasks of a queue classified with *DEFAULT* that can be performed in parallel),
- *`ThreadQueue.DEFAULT.queueCapacity`* (maximum capacity of a queue classified with *DEFAULT*),
- *`ThreadQueue.BOUNDED.maxRunning`* (maximum number of tasks of a queue classified with *BOUNDED* that can be performed in parallel) and
- *`ThreadQueue.BOUNDED.queueCapacity`* (capacity of a queue classified with *BOUNDED*)

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

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

❗ *The value for `ThreadQueue.<name>.maxRunning` must be smaller than the value for `ThreadPool.maxSize` (**Thread Pool** area, see above).*

❗ *For further information on these parameters, see "FirstSpirit Manual for Administrators", "Configuration of the FirstSpirit Server (fs-server.conf)" chapter, sub-section "Thread Pool Area" and "Thread Queues Area".*

Equally, the FirstSpirit security has been further increased.

❗ *fs4root must be updated in order for this change to take effect. If the update is not made, Client and Server cannot connect over HTTP.*

6.5 Innovations in the licence management

Two new licence parameters have been introduced in Version 4.2R4:

6.5.1 license.OFFICE_INTEGRATION

This is the licence parameter for the office integration. It must have the value `1`, so that Microsoft Office, OpenOffice or Google Docs text documents can be used in the FirstSpirit AppCenter (`license.OFFICE_INTEGRATION=1`). Without a valid licence, the tab for office engine configuration is greyed out in the project properties (see Chapter 6.2.3 page 129), the "View" \ "Office Engine" menu is deactivated in JavaClient.

6.5.2 license.APPTAB_SLOTS

This is the licence parameter for the FirstSpirit AppCenter (see also Chapter 3 page 16). It indicates how many different AppCenter applications (i.e. customer's own applications, which access the Application-API) can be used. With `license.APPTAB_SLOTS=5`, for example, five different applications or URLs can be opened. It does not matter which applications these are. Because unlike licensing of a FirstSpirit (module) add-on, it is not the function that is licensed here, but the number of integrated applications. Any number of tabs can be opened in the AppCenter for each application licensed via this parameter. In the case of web applications, one separate licence is required for each integrated web service. The URLs used by an AppCenter application are displayed in the AppCenter licenses overview (see Figure 6-16) in order to be able to find out which web applications have been integrated. Here it must be noted that only URLs of an application which "logically" belong to the integrated web application may be used, and not those of a completely different web service.

The first client opened in an AppCenter application uses a licence for this application (the application is "registered") and increases the licence parameter counter by 1. The registration continues to exist even after the client has been exited. If the value of the `license.APPTAB_SLOTS` parameter has been reached, one other application can be started in the AppCenter for test and demo purposes (a corresponding warning is displayed in the JavaClient). No other applications can be started, however, the already registered applications can be opened in other tabs or in other clients.

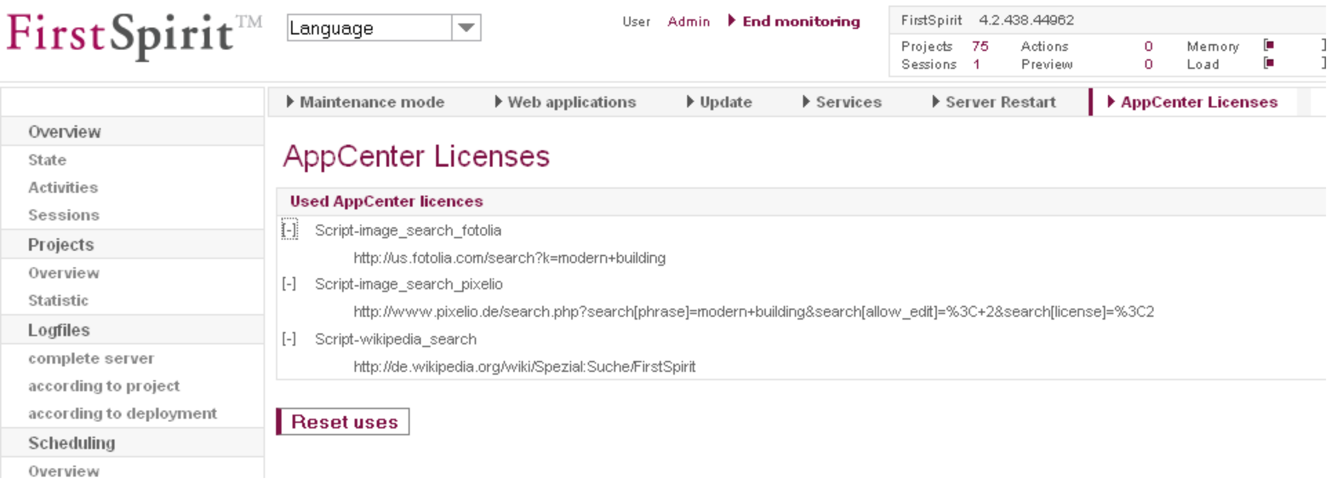
Several applications, which are displayed in the AppCenter, but are delivered as a standard with the FirstSpirit core product, are not covered by the licensing parameter `license.APPTAB_SLOTS` and are not counted as an AppCenter application, at present these are

- integrated browsers
- Preview of media (see Chapter 4.2.2.1 page 34)
- FirstSpirit Online Help (see Chapter 4.2.2.4 page 39)
- Enhanced image editing (see Chapter 4.2.3 page 40)
- Office integration (see Chapter 6.2.3 page 129) IMPORTANT: Office integration requires its own licence parameter, see Chapter 6.5.1 page 136.

The new input component FS_BUTTON provides a simple option for the integrating your own AppCenter applications (see Chapter 5.4 page 99). If this input component is used, each script and / or each module referenced by FS_BUTTON and a separate tab opened in the AppCenter, are counted as an application, which must be licensed via the parameter `license.APPTAB_SLOTS`.

The integrated preview is not an AppCenter application requiring licensing as it is a standard FirstSpirit function. An additional AppCenter licence is only required for the integrated preview if a separate enhancement or add-on is to be integrated in the FirstSpirit preview using AppCenter technology. Example: FS_BUTTON in the preview.

How many and which applications, which require licensing via the parameter `license.APPTAB_SLOTS`, are currently registered, can be checked in the Server Monitoring in the "AppCenter Licenses" submenu below the "FirstSpirit" / "Control" menu:



AppCenter Licenses

Used AppCenter licences

[-] Script-image_search_fotolia
http://us.fotolia.com/search?k=modern+building

[-] Script-image_search_pixelio
http://www.pixelio.de/search.php?search[phrase]=modern+building&search[allow_edit]=%3C+2&search[license]=%3C2

[-] Script-wikipedia_search
http://de.wikipedia.org/wiki/Spezial:Suche/FirstSpirit

Figure 6-16: Server Monitoring – AppCenter Licences

If necessary, the "Reset uses" button can be used to reset the number of registered applications to 0. Registered applications, which are currently open in Clients, can continue to be used until the application or the corresponding tab is closed. The server does not have to be restarted.

6.6 NTLM authentication and Windows 7

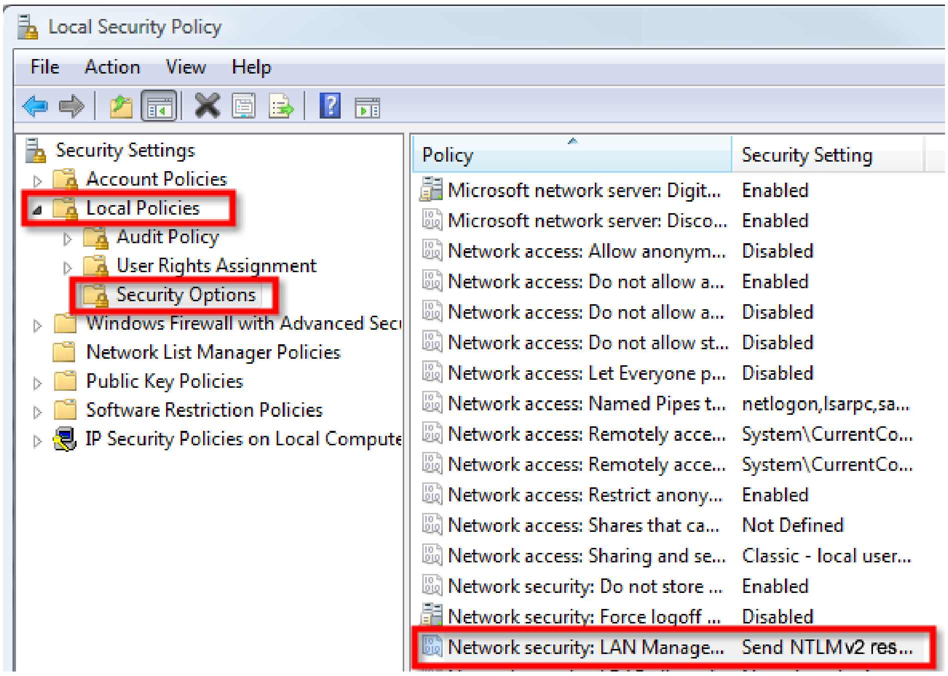
The NTLMv2 method is used as a default for authentication in the operating systems Windows Vista, Windows 7 and Windows Server 2008 R2.

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

> **!** *The NTLMv2 method is the successor of NTLM(v1) and has been developed due to security problems. Use of NTLM(v1) should therefore be carefully considered.*

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

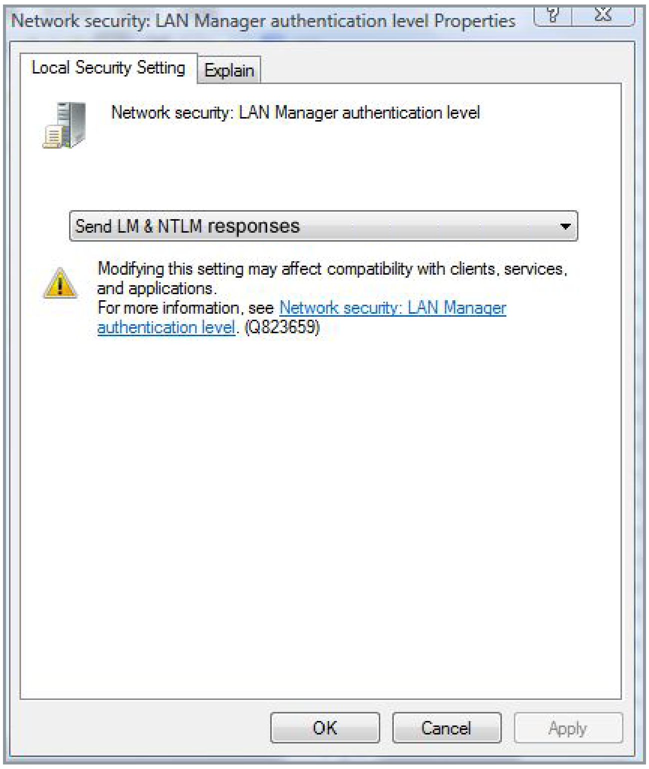
1. Press <Windows key> + <R>
2. enter `secpol.msc` and press <Enter>
3. Switch to "Local Policies" / "Security Options":



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

Figure 6-17: Network security: LAN manager authentication level

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


Network security: LAN Manager authentication level Properties
Local Security Setting
Explain
Network security: LAN Manager authentication level
OK
Cancel
Apply

Send LM & NTLM responses

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

Figure 6-18: Default setting in Windows XP

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

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

6.7 WebEdit-Themes

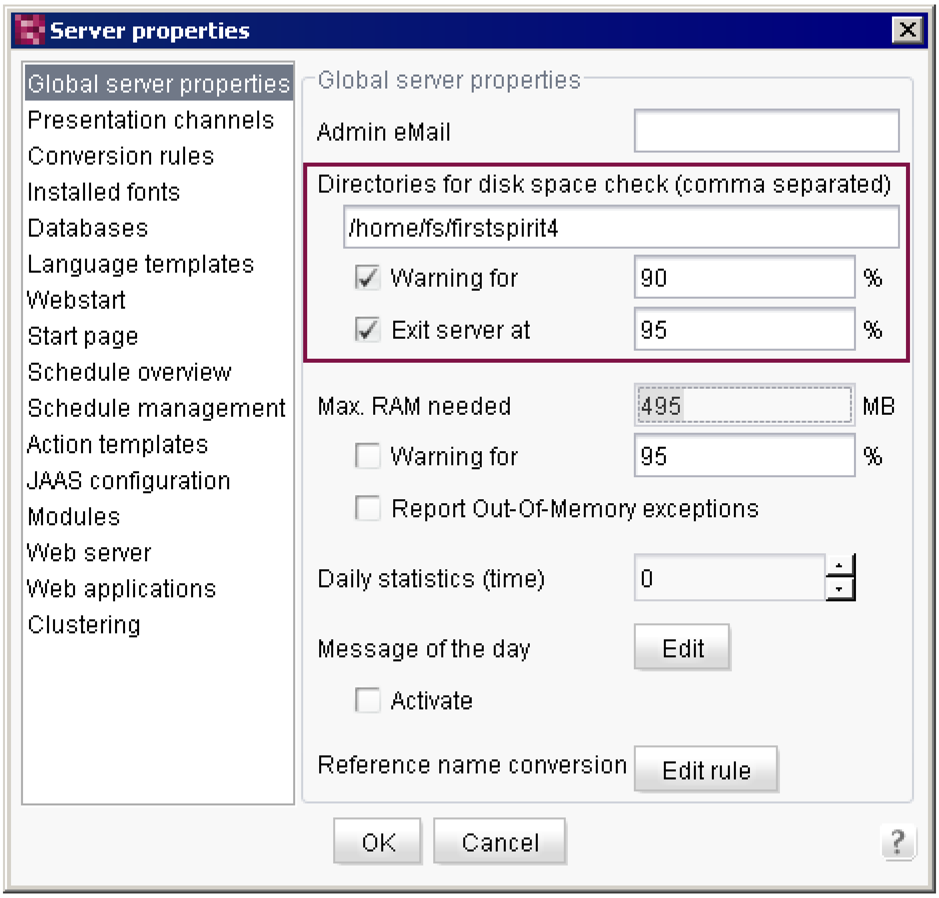
From 4.2R4, only the "xp" theme will be supported. This is now the default theme. In addition, themes with customer-specific settings will no longer be supported. The "WEBedit Theme" combobox in the Project properties (application for Server and Project Configuration / "Project" / "Webedit settings" area) is no longer available from 4.2R4.

6.8 Controlling disk space

With version 4.2R4 several measures have been implemented for being able to better control already used or still free disk space and thus to avoid possible data loss.

For this purpose, directories can be specified in the configuration file `fs-server.conf` or in the server properties (application for "Server and Project Configuration", menu "Server" / "Properties") under "Global server properties", which are to be controlled, and threshold levels for a warning of the server administrator via e-mail and the shut down of the FirstSpirit server. The free disk space in the specified directories is checked within an interval of five minutes.

For technical reasons this controlling of the disk space is only possible with JDK 1.6.



Global server properties

Admin eMail

Directories for disk space check (comma separated): /home/fs/firstspirit4

☑ Warning for 90 %

☑ Exit server at 95 %

Max. RAM needed 495 MB

□ Warning for 95 %

□ Report Out-Of-Memory exceptions

Daily statistics (time) 0

Message of the day Edit

□ Activate

Reference name conversion Edit rule

Figure 6-19: Server properties – Global server properties

Directories for disk space check (comma separated): Enter the directories in this field which are to be controlled regarding the disk space which is still available. By default, the base directory of the FirstSpirit server is specified. If FirstSpirit is distributed across several mount points, all mount points of the file systems which are used by FirstSpirit, must be specified here. If there is more than one directory, they must be separated by comma.

Warning for: Enter the percentage in this field, from which a warning e-mail is to be sent to the server administrator. Subject and content of the e-mail: "FIRSTspirit server 'MYSERVER' disk space warning: /home/fs/firstspirit4, 32,01 GB free, 42,52 GB used".
The percentage results from the relation of the disk space of the indicated directory which is still free and which is already occupied. By default, the value is set to 90%, i.e. the server administrator will get a warning e-mail if 90% of the disk space available are occupied (see also below, parameter `hdd.limit`). If more than one directory is specified, an e-mail will be sent if the percentage in **one** of the directories is exceeded. The value you enter here should be lower than the value for "Exit server at" (see below). Warning e-mails are sent every 12 hours maximal.

Exit server at: Enter the percentage in this field, at which the FirstSpirit server is to be shut down and an e-mail is to be sent to the server administrator. Subject and content of the e-mail: "FIRSTspirit server 'MYSERVER' disk space shutdown limit reached: /home/fs/firstspirit4, 463.2 MB free, 5.54 GB used".
The percentage results from the relation of the disk space of the indicated directory which is still free and which is already occupied. By default, the value is set to 95%, i.e. the server will be shut down if 95% of the disk space available are occupied (see also below, parameter `hdd.shutdown`). If more than one directory is specified, the server will be shut down, if the percentage in **one** of the directories is exceeded. The value you enter here should be higher than the value for "Warning for" (see above).

You can also specify the directories which are to be controlled and the threshold levels directly in the configuration file `fs-server.conf` and via the Server-Monitoring ("FirstSpirit" / "Configuration" / "Server"):

`hdd.directories`: Use this parameter to indicated the directories which are to be controlled. Default value is the FirstSpirit base directory, i.e. `hdd.directories=${cmsroot}`. This will be used if no value is given explicitly.

`hdd.limit`: Use this parameter to indicated the percentage from which a warning e-mail is to be sent to the server administrator. You can use values between `1` and `99`. If no warning e-mails are to be sent, the value can

be set to `-1`.
Default value is 90%, i.e. `hdd.limit=90`. This will be used if no value is given explicitly or if the given value is not between `1` and `99` or `-1`.

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

Corresponding warnings will be recorded in the log file `fs-server.log` in case of missing or invalid configuration, e.g.

- no or invalid directories:

```
WARN  22.03.2011 14:53:47.804
(de.espirit.firstspirit.server.ServerManagerImpl): Empty fs-
server.conf property: hdd.directories
```

or

```
WARN  22.03.2011 14:53:47.804
(de.espirit.firstspirit.server.ServerManagerImpl): Illegal fs-
server.conf property: hdd.directories, illegal directory: C:\test
```

- no or invalid threshold levels:

```
WARN  22.03.2011 14:53:47.805
(de.espirit.firstspirit.server.ServerManagerImpl): Illegal fs-
server.conf property value: hdd.limit

WARN  22.03.2011 14:53:47.805
(de.espirit.firstspirit.server.ServerManagerImpl): Illegal fs-
server.conf property value: hdd.shutdown=120
```

- no e-mail address:

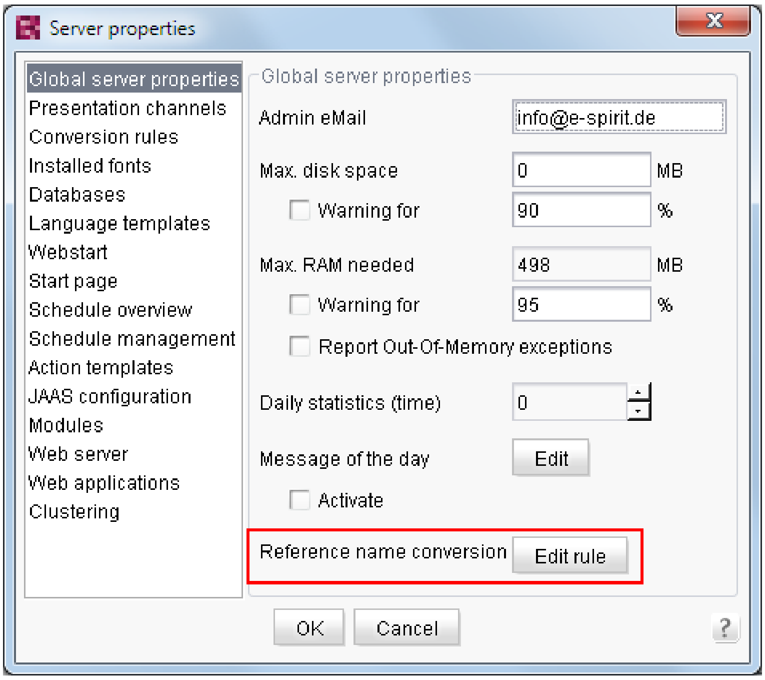
```
WARN  22.03.2011 15:11:30.739
(de.espirit.firstspirit.server.ServerManagerImpl): Empty fs-
server.conf property: mail.default-recipient
```

6.9 Reference names: Automatic conversion of special characters

Reference names may not contain special characters in any store, in FirstSpirit JavaClient; upper case (capital) letters must also not be used for reference names. In JavaClient, these invalid characters are removed during the input, upper case letters are converted to lower case letters. In FirstSpirit WebClient, special characters are automatically converted into underscores (_) and upper case letters are also allowed.

With FirstSpirit Version 4.2R4, a server-wide rule set can be defined, according to which invalid characters are to be transformed on creating new FirstSpirit objects or changing the reference name (using the context menu "Extras" / "Change reference name"). In this way, e.g. each "Ä" can be automatically transformed into an "ae" on input or special characters such as dots or commas transformed into underscores. This is useful, e.g. if reference names are used in the generation of a project for the generation of URLs (see also Chapter 4.1.3 page 28).

To this end, the "Edit rule" button is clicked in the Server Properties (application for Server and Project Configuration, "Server" menu, "Properties" entry) under "Global server properties":



Server properties

Global server properties

Admin eMail: info@e-spirit.de

Max. disk space: 0 MB

□ Warning for: 90 %

Max. RAM needed: 498 MB

□ Warning for: 95 %

□ Report Out-Of-Memory exceptions

Daily statistics (time): 0

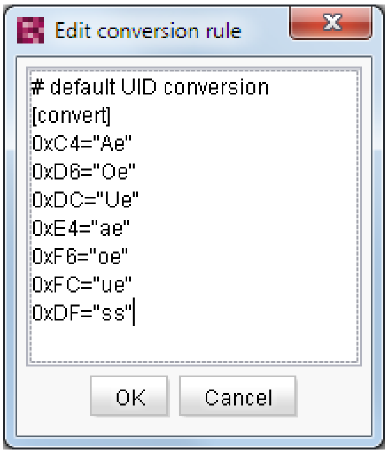
Message of the day

□ Activate

Reference name conversion

Figure 6-20: Server properties – Global server properties

The following window opens:



```
# default UID conversion
[convert]
0xC4="Ae"
0xD6="Oe"
0xDC="Ue"
0xE4="ae"
0xF6="oe"
0xFC="ue"
0xDF="ss"
```

Figure 6-21: Reference name conversion

Several rules are already defined here as a default, namely for the umlauts "ä", "ö" and "ü" in lower and upper case and for "ß". Each rule must be positioned in a separate line and consists of two values, separated by an equals sign:

- **on the left** the special character to be transformed, in ASCII code (hexadecimal)
- **on the right** the valid character(s), into which the special character is to be transformed when used in reference names, in double inverted commas.

The default rule set transforms umlauts used in reference names into 2 characters (vowel in lower case + "e"), "ß" is transformed into "ss".

If the characters in this dialog are not given in the correct code or with the correct formatting, an error message is displayed in the following form when the rule set is saved:
"The conversion rule format is incorrect: Error parsing line 11:...".

The rules supplied can be changed or deleted and new rules can be added. They are not reset or supplemented during a FirstSpirit update.

Click "OK" to save the rules. If the dialog below it with the Server properties is closed with "OK", the rules are immediately applied; the server or project do not have to be restarted first.

6.9.1 Configuration

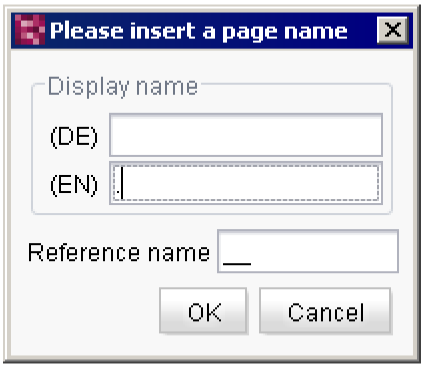
Rules can only be defined for single characters, not for character strings, and only language-independent, i.e. each rule applies to all languages. If more than one definition is given for a special character (ASCII code), the lowest definition is always applied. Special characters for which no rule is deposited continue to be removed directly on entering a reference name.

If upper case letters are defined on the right-hand side, they are automatically transformed into lower case letters in JavaClient.

If a definition on the right-hand side contains (special) characters, for which in turn a conversion rule is defined, these (special) characters are replaced accordingly in the "Display name" field (cf. Figure 6-22), e.g. in the following definition (with spaces at 0x2E)

```
0x2E="_ "
0x20="_"
```

a dot . is converted into a double underscore:



(EN) .

Reference name __

Figure 6-22: Transformed reference name

In the "Reference name" field and / or if no definition exists for the special character, it is removed directly on entry.

In addition, comments can be inserted, which have to be introduced with #.

6.9.2 Symbols and equivalent ASCII coding (hexadecimal)

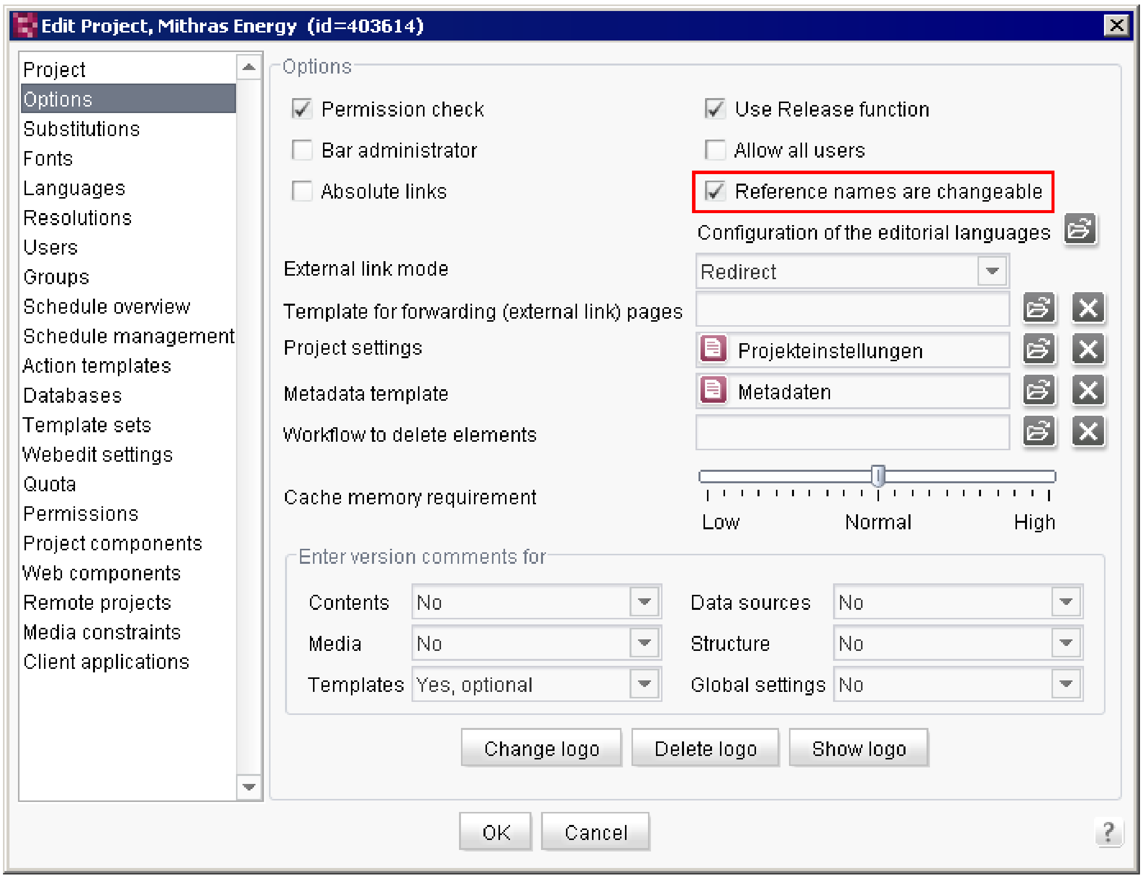
Name	Symbol	Hexadecimal value
spaces		0x20
exclamation mark	!	0x21
quotation marks	"	0x22
hash	#	0x23
dollar sign	$	0x24
percent sign	%	0x25
ampersand	&	0x26
apostrophe	,	0x27
left parenthesis	(	0x28
right parenthesis	)	0x29
asterisk	*	0x2A
plus sign	+	0x2B
comma	,	0x2C
hyphen	-	0x2D
period	.	0x2E
semicolon	;	0x3B
colon	:	0x3A
less-than	<	0x3C
equals-to	=	0x3D
greater-than	>	0x3E

Name	Symbol	Hexadecimal value
question mark	?	0x3F
slash	/	0x2F
at sign	@	0x40
underscore	_	0x5F

> ! *When defining conversion rules for the symbol . (dot), it should be taken into account that it is used in FirstSpirit to generate reference names for table and link templates and is also converted if a corresponding rule definition exists.*

6.10 Reference names: Prevent change project-wide

Since FirstSpirit Version 4.1.24, reference names can be changed in JavaClient using the context menu "Extras" / "Change reference names" by each editor who has the "Change" right on the object concerned. With the launch of FirstSpirit Version 4.2R4, the administrator can prevent this option project-wide in the Project properties, namely in the "Options" area:



Options

☑ Permission check
☑ Use Release function
☐ Bar administrator
☐ Allow all users
☐ Absolute links
☑ Reference names are changeable

Configuration of the editorial languages

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

Enter version comments for

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

Figure 6-23: Project properties – Reference names

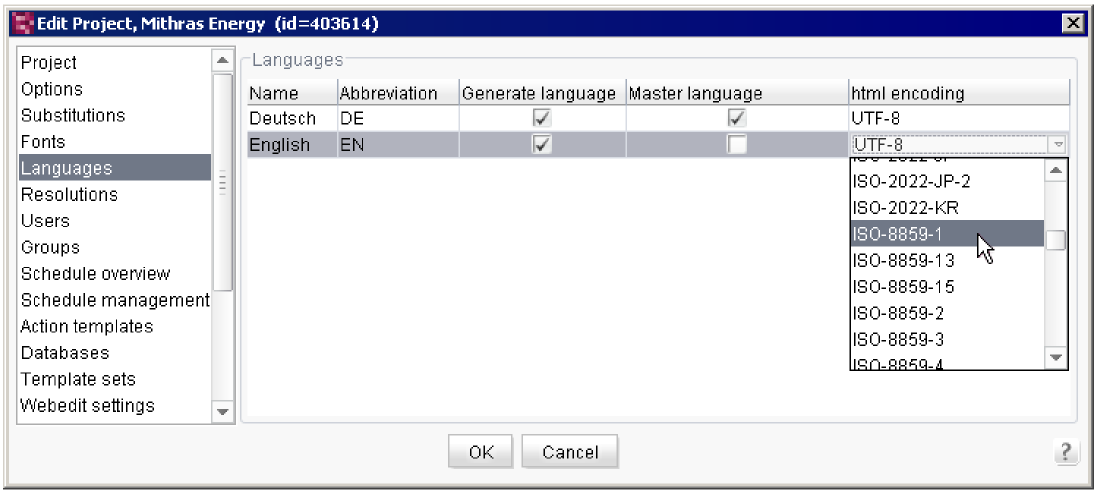
If the "Reference names are changeable" checkbox is activated, reference names can be changed in the project concerned. If the checkbox is deactivated, reference names cannot be changed, the "Change reference names" menu item is greyed out. As a default, the option is enabled when creating new projects, so that reference names can be changed as to date. Regardless of the setting in the Project properties, server and project administrators can change reference names at any time.

The "Rename" function (F9) in the JavaClient is not affected by the configuration of the "Reference names changeable" option in the Project properties: Reference names of elements without UID (e.g. sections in the Page Store) can continue to be changed using "Rename". Changes to the reference names of sections can, however, be prevented by disabling the "Display reference names" option in the Project properties, "Options" area, "Configuration of the editorial languages". A change is then only possible via BeanShell (API).

6.11 Simplified encoding definition for languages

In the Project properties, HTML encoding can be defined for each project language under the "Languages" item, e.g. ISO-8859-1 for West European languages. This was entered so far directly in the "html encoding" field. The input had to comply with a whitelist maintained by FirstSpirit, encodings which are not supported were ignored, i.e. the previously saved encoding was always used.

With 4.2R4, the "html encoding" text field has been changed into a combobox. All encodings supported by the respective Java version under which the FirstSpirit Server runs are now displayed in this combobox. Click the field to open the combobox, and the required encoding can be selected:



Name	Abbreviation	Generate language	Master language	html encoding
Deutsch	DE	☑	☑	UTF-8
English	EN	☑	□	UTF-8

Figure 6-24: Project properties – Languages – Encoding

If, on changing from one JDK version to another, the encoding is not supported (any more), this is visualised with red lettering. If a project is exported from one FirstSpirit Server to another, which does not support an encoding used in the project, a warning is issued during the import in the form

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

6.12 New functions for workflows and tasks

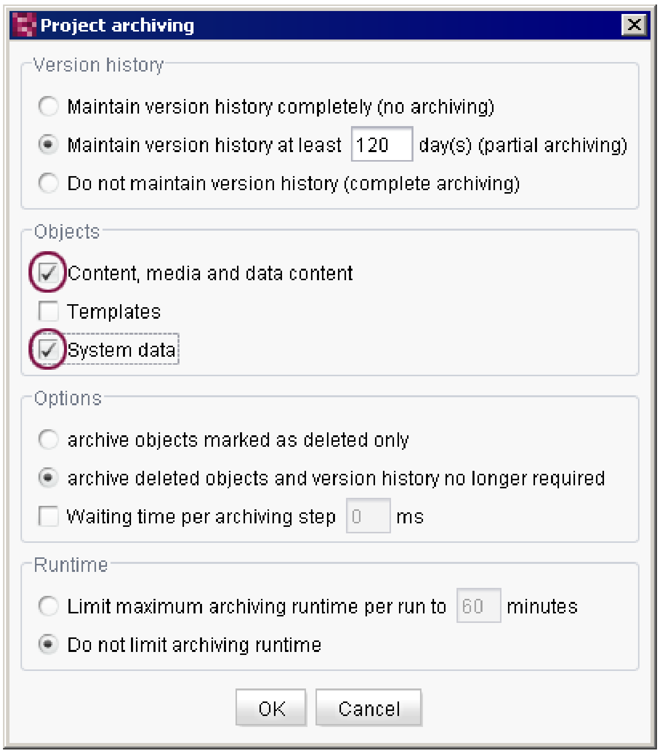
The following new functions have been implemented in order to increase performance, especially in projects with many tasks:

6.12.1 Archiving closed tasks

The "Archive old project states" schedule can be used to swap out data from the project which is no longer required, in order to reduce loading times and to increase the performance of the FirstSpirit Servers. To this end, data is moved from the repository of a project (in the server directory `data\projects\`) into archive files. It is possible to define which type of data is to be archived (see also *FirstSpirit Manual for Administrators*, "Archive old project states" chapter):

- **Content, media and data content**: this option is used to archive all content of the Page, Media and Content Store of a project.
- **Templates**: this option is used to archive all content of the Template Store.
- **System data**: System data is information which is generated by the system with each action in JavaClient (e.g. create or delete objects, releases, etc.). If this option is selected, system data which is no longer used is archived. The system data also includes revisions.

From FirstSpirit Version 4.2R4, closed tasks are now also archived. To do this, the "Page, Media and Content" and "System Data" checkboxes must be activated:



Project archiving

Version history

○ Maintain version history completely (no archiving)

● Maintain version history at least 120 day(s) (partial archiving)

○ Do not maintain version history (complete archiving)

Objects

☑ Content, media and data content

□ Templates

☑ System data

Options

○ archive objects marked as deleted only

● archive deleted objects and version history no longer required

□ Waiting time per archiving step 0 ms

Runtime

○ Limit maximum archiving runtime per run to 60 minutes

● Do not limit archiving runtime

Figure 6-25: Archiving closed tasks

In this case, all files belonging to a task which have been closed by a certain time lying within the period to be archived are archived.

6.12.2 Configure cache memory for workflows and tasks

In the "Server" area of the configuration file `fs-server.conf`, the `workflow.model.cache` and `workflow.task.cache` parameters can be used to set the cache size and type for workflow models and tasks. The type which is defined here provides information about how long workflow models or tasks remain in the cache. A distinction is made between `WEAK` and `SOFT`: if `WEAK` is indicated, the objects will be deleted from the cache as soon as they are no more used. If `SOFT` is indicated, the objects will, depending on the used VM, remain in the cache until there is insufficient memory space. (The type `WEAK` is usually advantageous for a large memory.) The size of the LRU in KB must be attached to the type, separated by an underscore.

The following values are defined as a default:

```
workflow.task.cache=SOFT_1024
workflow.model.cache=SOFT_128
```

6.12.3 Sending workflow e-mails to groups

There is now an option in the project properties for specifying an e-mail distribution list for groups, to which the notification e-mails are sent when a workflow activity or transition is performed (see *FirstSpirit Manual for Developers (Part 1: Basics)*, subsection "E-mail tab" in the chapters "Properties of an activity" and "Properties of a transition"). This is defined in the "Groups" area of the project properties (see also Figure 6-29). In this way it is now possible to send e-mails, which are sent as part of a workflow, to be sent to all members of external groups.

The dialog on creating a group has been extended to include a corresponding field:

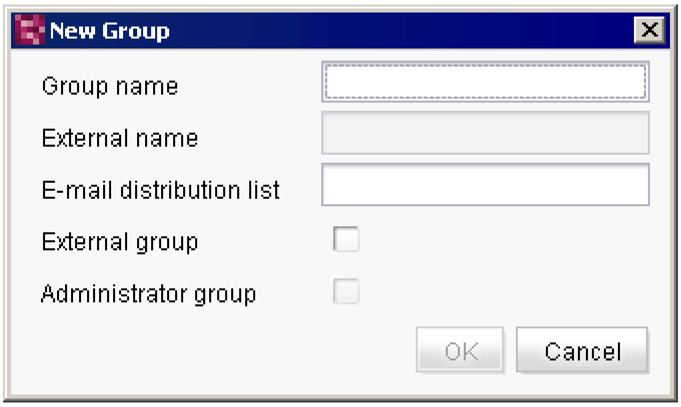


Figure 6-26: Create new group

To add an e-mail distribution list to an existing group, right-click the required group to open the context menu. Select the "Edit e-mail distribution list" entry:

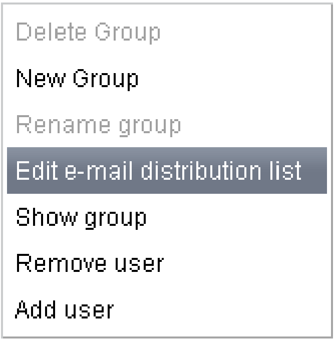


Figure 6-27: Edit e-mail distribution list

The required e-mail address is entered in the dialog that opens:

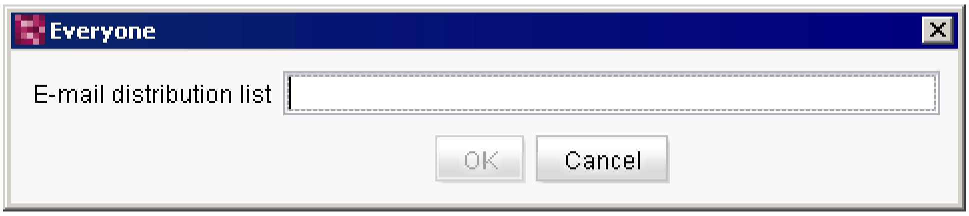
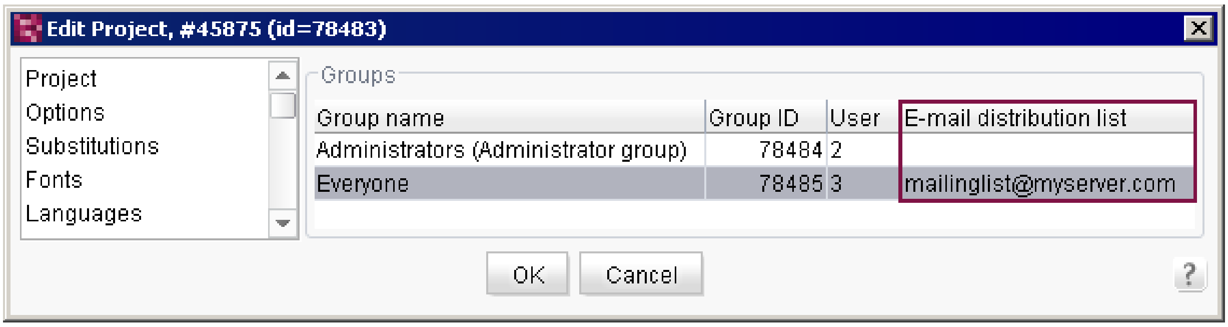


Figure 6-28: Specify e-mail distribution list

Several e-mail addresses can be entered, separated by semicolons. The field can also be left empty, or e-mail addresses already entered can be deleted.

Press "OK" to accept the changes for the group, the e-mail distribution list is now displayed in the groups overview:



Group name	Group ID	User	E-mail distribution list
Administrators (Administrator group)	78484	2	
Everyone	78485	3	mailinglist@myserver.com

Figure 6-29: Project properties – e-mail distribution list for groups

The workflow e-mails are then also sent to the e-mail addresses defined here. If no e-mail address is given for a group, the e-mails are sent as to date to all users of the group, however, employees of external group potentially do not receive any e-mails.

7 New/Modified Functions in Modules

On updating to 4.2R4, all installed modules must be updated, as a new signing key is used in 4.2R4 and otherwise error messages can occur (see FirstSpirit Manual for Administrators, "Modules" chapter).

At this point it should also be noted that in the database context from 4.2R4, integration of JDBC drivers via modules is recommended. See also Chapter 6.1 page 111.

The installation or update of modules should only be executed within "maintenance slots" because otherwise for example, a running generation can lead to errors (e.g. NullPointerException).

7.1 FirstSpirit Portal SAP EP BP

For detailed information on the FirstSpirit Portal SAP EP BP module, please also refer to the relevant module documentation.

7.1.1 Evaluation of groups in CMS_INPUT_PERMISSION

The "FirstSpirit Portal SAP EP BP" module can be used to realise role-dependent navigation. These roles are assigned using the CMS_INPUT_PERMISSION input component, which enables a hierarchical definition of groups. The settings made here are transferred into the XML file for the navigation, which in turn provides the basis for navigation in the portal. Each role in this XML file is created in the project and is assigned its own navigation in the Site Store.

Until now, only the authorised groups on the lowest level, i.e. without sub-groups, were taken into account. With 4.2R4, the authorised groups of **all** levels are now taken into account, including those which may only have been used as a structuring aid.

From this, the result is that additional groups or roles and additional navigation nodes can be added in the Site Store in all projects which use this module.

7.1.2 Modification of the "Portal_createLink"

The script for link generation, "Portal_createLink", has been changed with Version 4.2R4 and the `CreatePortalLink` class has been extended so that it directly implements the interface to avoid calling the Beanshell interpreter.

Following an update of the "FirstSpirit Portal SAP EP BP" module, the call of the "Portal_createLink" script should be changed over as follows in all projects concerned:

```
#!executable-class
de.espirit.firstspirit.opt.portal.script.CreatePortalLinkExecutor
```

7.2 FirstSpirit Personalisation

In order to prevent Open-Redirect attacks, external redirects are now prohibited in projects which use the FirstSpirit Personalisation module. I.e. only relative and absolute URLs are possible (e.g. *start.jsp*, *../area/index.html*, *../area/search.jsp*). Exception: Redirects to the same host or a host in the same domain are allowed. A forwarding (redirect) page must be created for redirecting to an external URL.

7.2.1 User login via Kerberos

From FirstSpirit Version 4.2R4 login via Kerberos in combination with the module FirstSpirit Personalisation is possible. For this reason, the new login module "Kerberos Login“ has been implemented.

For detailed information about the configuration of the login module please see module documentation "FirstSpirit Personalisation".

7.3 FirstSpirit Search

In order to prevent Open-Redirect attacks, external redirects are now prohibited in projects which use the FirstSpirit Search module. I.e. only relative and absolute URLs are possible (e.g. *start.jsp*, *../area/index.html*, *../area/search.jsp*). Exception: Redirects to the same host or a host in the same domain are allowed. A forwarding (redirect) page must be created for redirecting to an external URL.

! *For detailed information on the FirstSpirit Search module, please also refer to the relevant module documentation.*

7.4 FirstSpirit PackagePool

Using the FirstSpirit PackagePool you can transfer objects from a (master) project into one or more target project/s, so that content can be reused. By updating the packages, modifications which have been effectuated in the master project and which are part of a package can be assumed easily into the target projects.

In many cases the order in which the objects are existing in the master project, within a chain of objects, plays a decisive role. The order from the master project should be retained as much exactly in the target project when importing packages initially or when importing packages with modifications and also in cases in which chains of objects are enhanced in the target project. The following modifications have been implemented in this context in Version 4.2R4, with the intention to retain the order of imported objects within a chain of objects even more logically and more stringently:

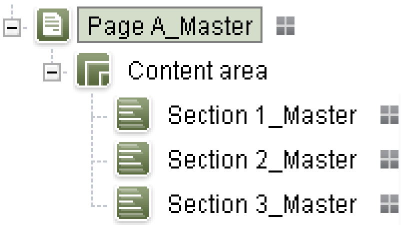
When inserting objects from the master project into the target project, the relation to the precedent object ("parent") is preferential. I.e.,

- if both a parent and a child exist, the new object will be inserted behind the parent,
- if only a parent exists, the object will be inserted behind the parent,
- if only a child exists, the object will be inserted before the child

Previously, objects was tied more strongly to the following object ("child"), as far as a child was existing.

Example 1:

The master project contains a page with 3 sections in the following order:

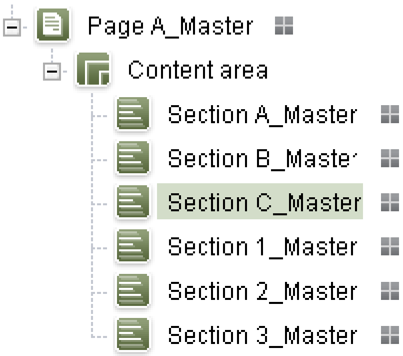


- Page A_Master
  - Content area
    - Section 1_Master
    - Section 2_Master
    - Section 3_Master

Figure 7-1: PackagePool – Page with three sections

If a package with these objects is imported into the target project, the order of the sections remains.

Now, 3 further sections are added to this page, namely **before** the sections which are already existing:



- Page A_Master
  - Content area
    - Section A_Master
    - Section B_Maste
    - Section C_Master
    - Section 1_Master
    - Section 2_Master
    - Section 3_Master

Figure 7-2: PackagePool – New sections

Section C commits itself stronger to its parent, so that it will be inserted after Section B when it is imported, as it is expected.

Example 2: A section is inserted in the target project at the last position of a page with 3 sections which derive from the master project:

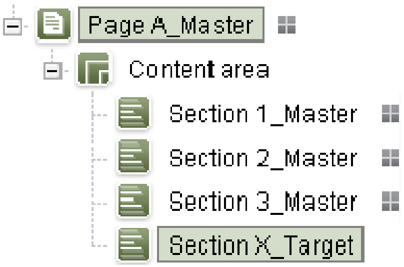


Figure 7-3: PackagePool – Combined sections in the target project

In the master project a section is inserted at the last position of the page as well:

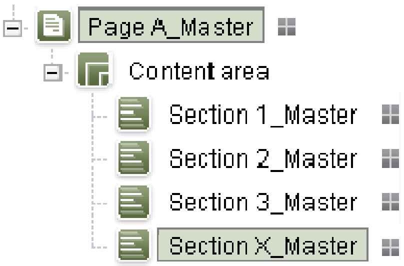


Figure 7-4: PackagePool – Master project

When the updated package is imported from the master into the target project, the newly inserted section from the master project will be inserted **before** the newly inserted section in the target project.

! *If sections are re-sorted in the master project and imported into the target project in which these sections are already existing, they will not be re-sorted in the target project. In order to assume the order of the sections from the master project, the respective page should be deleted in the target project and the package with the re-sorted sections should be imported into the target project again.*

8 Appendix

8.1 Renaming of FirstSpirit modules

The entire FirstSpirit module offer was restructured with effect from 01.01.2011. The names of the modules were changed and they have been assigned to so-called suites. The standardisation of the module names and categorisation is intended to make the existing over simpler and clearer. The changeover is purely organisational in nature, the functional scope of the modules does not change at all.

The following table shows the old and new names:

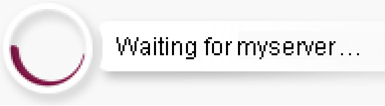
old	**new**	**File name of the module**
Multisite Management Suite		
Package Pool	CorporateContent	
Database Pool	CorporateDatabase	
Remote Media	CorporateMedia	
Infrastructure Suite		
HighAvailability	HighAvailability	
EnterpriseBackup	EnterpriseBackup	
Preview Server	PreviewServer	
Generation Server	Generation Server	
API Access	API-Access	
Integration	Dynamic Database Access	*fs-integration.fsm*
Information Suite		
Backend Suite		
EnterpriseConnect	EnterpriseConnect	*fs-ep-connect.fm*

old	new	File name of the module
DMS-Connect	DMS-Connect	
PIM Integration	PIM-Connect	
Translation Connect	TranslationConnect	
Office	OfficeConnect	*fs-office.fsm*
Author Experience Suite		
	OfficeIntegration (from 4.2R4)	
Workflow	WorkflowDesigner	
Newsletter	EmailMarketing	
	WebControlling (from 4.2R4)	
	VideoManagement (from 4.2R4)	
Portal Suite		
SAP Business Package / Portal SAP EP BP	SAP Business Package for FirstSpirit	*fs-portal.fsm*
IBM Websphere Portal	IBM Websphere Portal Integration	
	Liferay Portal Integration (ab 4.2R4)	
Customer Experience Suite		
Personalisation	DynamicPersonalization	*fs-perso.fsm*
	SocialWeb	
Search	BasicSearch	*fs-search.fsm*
Exalead	EnterpriseSearch	
Publishing Suite		
PDF Professional	PDF-Publisher	

old	new	File name of the module
Mobilisation	MobilePublisher	

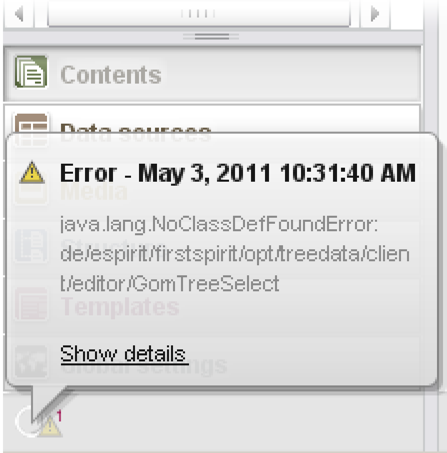
8.2 Changes to software behaviour

- **Layout – new loading icon and error visualisation:** A new loading icon is used in many places in FirstSpirit JavaClient 4.2R4, e.g. in the Integrated preview:

  Waiting for myserver...

  In addition, the preview is displayed faded.
  This icon is also used in the bottom left-hand corner, in which error messages that occur are also displayed. If an error message occurs, it is now displayed in a kind of speech balloon, the load icon contains a number, which represents the number of error messages that exist:

  Error - May 3, 2011 10:31:40 AM

  java.lang.NoClassDefFoundError: de/espirit/firstspirit/opt/treedata/client/editor/GomTreeSelect

  Click "show details" to open the dialog with the complete error message/s as usual.
- **Expanding / reducing windows:** For easier usability, windows which open in JavaClient (e.g. the translation help) no longer have a maximise icon, instead, if necessary, they can be maximised using the F11 shortcut for Windows and Cmd + Shift + M for Mac OS.
- **Input components – Changes to language specifications:** The <*LANGINFO*> tag can be used to define language-dependent specifications for each input component, e.g. what the label of tinput component is to be in the different project languages (*label* parameter), which tooltip is to be displayed (*description* parameter) etc. For reasons of clarity, specifications which are identical in several languages are condensed. For example,

```
<LANGINFOS>
   <LANGINFO lang="*" label="Date"/>
   <LANGINFO lang="DE" label="Date"/>
   <LANGINFO lang="EN" label="Date"/>
</LANGINFOS>
```

when saved becomes

```
<LANGINFOS>
   <LANGINFO lang="*" label="Date"/>
</LANGINFOS>
```

Previously, language information was only removed if the values were the same in all languages, as in the example above. In addition, the *format*, *expression*, *ignore* and *length* parameters are not taken into account in the condensing. Since 4.2R4, these parameters are taken into account and language specifications are now also condensed within *<LANGINFOS>*, if they are the same in at least two languages, e.g.

```
<LANGINFOS>
   <LANGINFO lang="*" label="Date" format="dd.MM.yy"/>
   <LANGINFO lang="DE" label="Date" format="dd.MM.yy"/>
   <LANGINFO lang="EN" label="Date" format="MM/dd/yy"/>
</LANGINFOS>
```

when saved becomes

```
<LANGINFOS>
   <LANGINFO lang="*" label="Date" format="dd.MM.yy"/>
   <LANGINFO lang="EN" label="Date" format="MM/dd/yy"/>
</LANGINFOS>
```

The definition for the return value (*) is always retained, if specifications of languages have the same values, the top one is always retained. If the mandatory definition of the return value is missing, the first defined value is automatically used on saving, e.g.

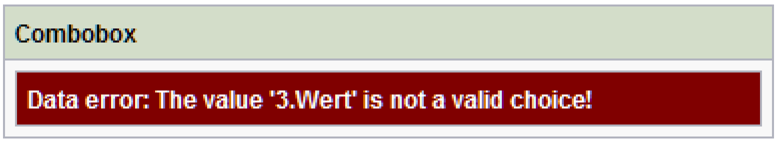
```
<LANGINFOS>
   <LANGINFO lang="EN" label="Date"/>
   <LANGINFO lang="DE" label="Datum"/>
</LANGINFOS>
```

when saved becomes

```
<LANGINFOS>
   <LANGINFO lang="*" label="Date"/>
   <LANGINFO lang="DE" label="Datum"/>
</LANGINFOS>
```

Another change concerns the *description* parameter, which can used to define a text which is to be displayed for the respective input component by means of the tooltip. If, from Version 4.2R4, *description* is not given, the text defined for the *label* parameter is displayed as the tooltip. Previously, no tooltip was displayed if *description* was not set.

- **CMS_INPUT_COMBOBOX – changed behaviour on deleting:** In the form of the CMS_INPUT_COMBOBOX input component, the *ENTRY* tag can be used to make entries available for selection by the editor. If a value has already been selected and saved in an input component in the Page or Content Store based on this form, and the relevant *ENTRY* definition has been deleted from the template, the following message is displayed when the input component is updated in the Page or Content Store:

  Combobox

  Data error: The value '3.Wert' is not a valid choice!

  Figure 8-1: Combobox with error message

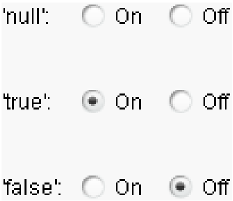
  In the event of renewed editing (e g. using Ctrl + E), a message is displayed in the form: "The data for the editor 'Combobox' is corrupt and cannot be displayed. Reason: Data error: The value '3. Wert' is not a valid choice! The editor value will be reset for further editing!". Click "OK" to remove the error message (Figure 8-1) from the input component, as well as the value previously saved. Another available value can now be selected.

  Previously, the saved value continued to be saved until a new value was selected.

  The same new behaviour has also been implemented for value sets, which are integrated in a combobox via CMS_INCLUDE_CONTENT or CMS_INCLUDE_OPTIONS: If a data record is deleted from a data source which was selected and saved in the combobox, the error message given above is also output and the value is reset on switching to edit mode. However, the deletion of data records which are referenced in a combobox is reserved for server and project administrators and is only possible after accepting a confirmation prompt.
- **CMS_INPUT_CONTENTLIST – adding data records:** If CMS_INPUT_CONTENTLIST is used in a table template which is connected to another table by an n:m relationship, data records would not be displayed as being changed if data records were added using this input component. If a data record was deleted, however, it was visualised accordingly. In Version 4.2R4, the behaviour has been changed so that now the data record is not displayed as changed, either on adding or on deleting. However, at present it is possible to visualise the change (adding or deleting) indirectly: to do this, a column of the type "xs:date" must be created in the table which is connected in the table template with a (hidden) CMS_INPUT_DATE input component. However, this procedure is not reliable, as the behaviour of the CMS_INPUT_DATE input

component could possibly also change in this respect in future FirstSpirit versions.

- **CMS_INPUT_NUMBER – formatting numbers:** Depending on the configuration by the template developer, numbers in the input component of the type CMS_INPUT_NUMBER can be displayed formatted according to the input. E.g. a comma (decimal point) and 2 zeros can be appended to a currency amount (the entry "12" becomes "12.00") or superfluous zeroes in the entry can be removed. From FirstSpirit Version 4.2R4, due to the refactoring measures (see Chapter 5 page 72), this formatting is not implemented until Edit mode is exited (e.g. using Ctrl + E), it was previously implemented on (interim) saving (e.g. using Ctrl + S).
- **CMS_INPUT_TOGGLE – modified visualisation of states:** If CMS_INPUT_TOGGLE is used with radio buttons (*type="RADIO"*), the input component has been displayed so far with the state *OFF*, until the state was changed by the editor to *ON*. However, no value was returned. From Version 4.2R4 the input component is displayed initially without any state. While the state *ON* returns the value *true* and *OFF* the value *false*, the newly introduced state returns *null* now:

  'null': ○ On ○ Off

  'true': ● On ○ Off

  'false': ○ On ● Off

  Figure 8-2: CMS_INPUT_TOGGLE

  The initial state (= no state) can not be restored by the editor after he changed it once.
- **FS_REFERENCE – folder display in WebEdit:** While, for successful selection options via FS_REFERENCE in WebEdit, previously higher level folders were not allowed to be hidden (e.g. using *HIDE*), so that it was possible to navigate to lower level elements, in 4.2R4, all folders needed to navigate to the required element are shown in the selection dialog. Therefore, with the following configuration, for example, all folders below the folder with reference name "powerinverter" are now displayed:

```
<FILTER>
      <ALLOW type="media"/>
      <HIDE type="mediafolder"/>
</FILTER>
...
<SOURCES>
      <FOLDER name="powerinverter" store="mediastore"/>
</SOURCES>
```

- **Resolution calculation of images in JPG format:** If resolutions for images are defined for a project (Project properties, "Resolutions" area), if necessary, all images in the Media Store are automatically scaled by the system to the sizes given for these resolutions and saved. Regardless of the file format of the original image (ORIGINAL resolution), these image files generated by the system are saved in PNG format. Exception: Original images in JPG format are also saved as JPG in the resolutions. With immediate effect, this procedure is also applied to images which are manually trimmed (JavaClient, Media Store: "Edit image cutout" function for Media,). Previously, manually trimmed images in JPG format were also saved in PNG format. In addition, following automatic scaling and manual trimming, image files with file extension JPEG are also saved in JPG format; in previous versions they were saved as a PNG file.

- **"Change reference names" function disabled project-wide:** Depending on the project configuration by the administrator, the context menu function "Extras" / "Change reference name" available on most FirstSpirit objects until now, can be disabled (except for server and project administrators, see Chapter 6.10 page 149):

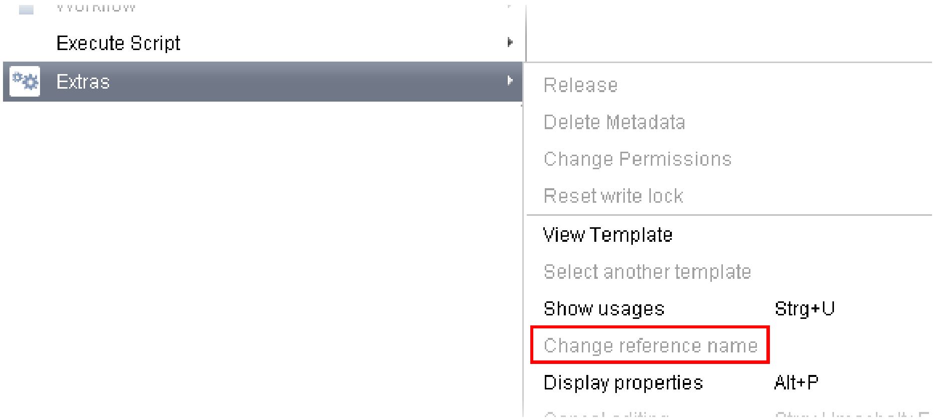
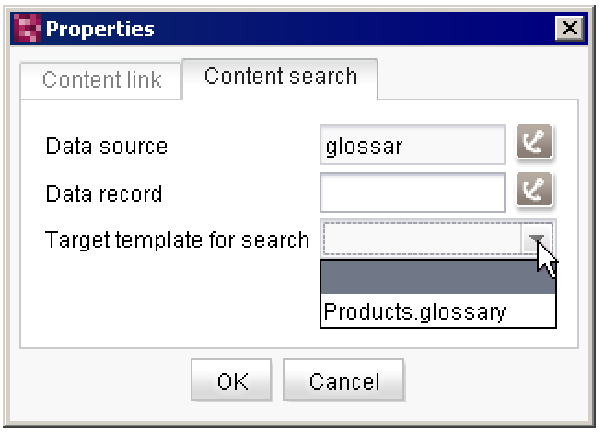


Figure 8-3: Project-wide disabled "Change reference name" function

- **Style templates: New "Properties" tab:** Style templates (special format templates in the Template Store), which are used for creating inline tables, now have a new "Properties" tab. The unique name (reference name) of the style template is displayed on this tab. In addition, default values can now be defined using the available "Default values" button.

- **"Convert link template":** Before the introduction of generic link editors, the three link types "internal link“, "external link" and "data link" were the only link types available in FirstSpirit. In a later version of FirstSpirit, the three link types: "internal link", "external link" and "data link" will be removed. For this reason, use of generic link editors is recommended. FirstSpirit provides a conversion function for these three link types to make the changeover to generic link editors easier. The conversion function is contained in the context menu of JavaClient under the "Extras" sub-item and is called "Convert link template". When entering data links, the three "Data source", "Data record" and "Target template for search" input fields can be edited:



Data source: glossar

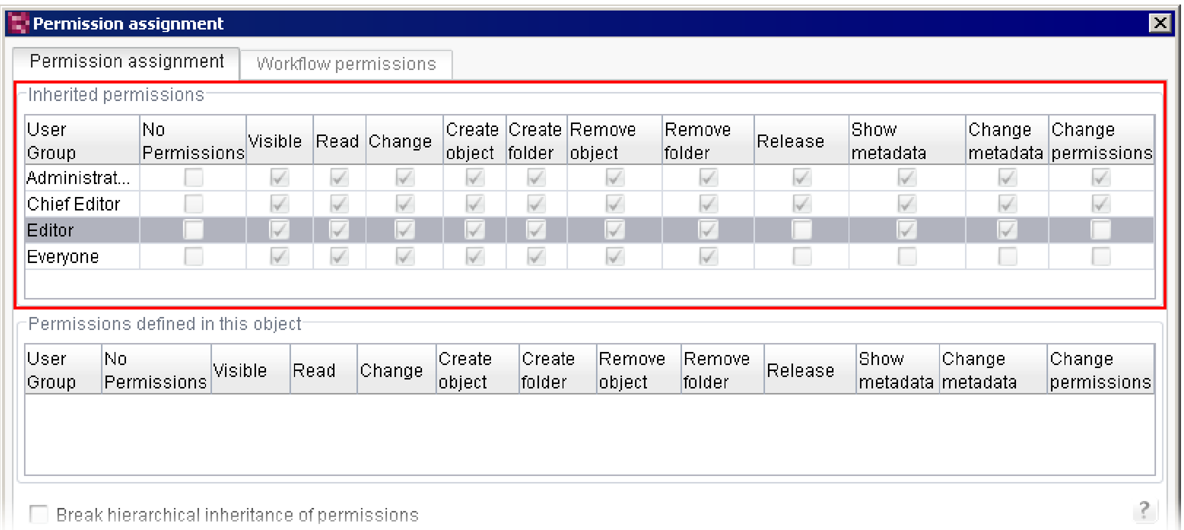
Data record:

Target template for search: Products.glossary

Figure 8-4: Data link – tab "search"

With these input fields it is possible to create a link to a specific data record, which is output with the selected table template. Following conversion of a data link, the three input fields are joined to form one input field called "Data record" (variable name "content"). Due to this joining, it is no longer possible to select from all table templates for a table in the schema (cf. FirstSpirit Online Documentation, "Template development" / "Link templates" / "Configuration" / "for data links" chapter, description of "sectiontemplate"). In the case of generic links, only the table template on which the data source is based, from which the data record was selected is returned. If different table templates to this table template are required, these data sources must be created in the Content Store and the data record selected from this data source. Alternatively, the relevant table template can also be given in the link template. In previous versions of the FirstSpirit 4 family, the "template" prefix was added before the reference name of the table template on outputting the `sectiontemplate` attribute with the `#link` system object. This prefix is now no longer used for the output, i.e. `#link.sectiontemplate` now only outputs the reference name of the table template. Use of the `#link` system object in generic links is not recommended. Instead, we recommend directly using the variable names of the individual input components.

- **Permission assignment dialog on root nodes:** The project administrator can assign users or user groups on every node of the tree structure. To do this, the permissions dialog is opened using the "Extras" / "Change Permissions" context menu. Among other things, the permissions inherited from higher nodes are displayed here:



Inherited permissions

User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release	Show metadata	Change metadata	Change permissions
Administrat...	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Chief Editor	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Editor	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☐
Everyone	☐	☑	☑	☑	☑	☑	☑	☑	☐	☐	☐	☐

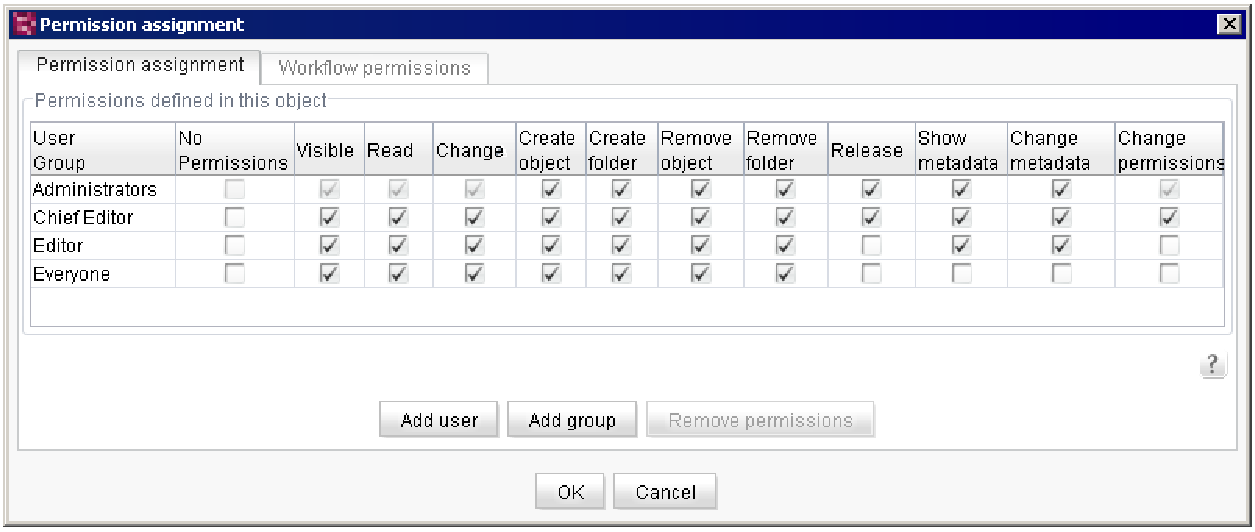
Permissions defined in this object

User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release	Show metadata	Change metadata	Change permissions

☐ Break hierarchical inheritance of permissions

Figure 8-5: Permission assignment dialog – Inherited permissions

Previously, the "Inherited permissions" area was also displayed at root node level, however it was "empty". The "Permissions defined in this object" area only is now displayed on root nodes:



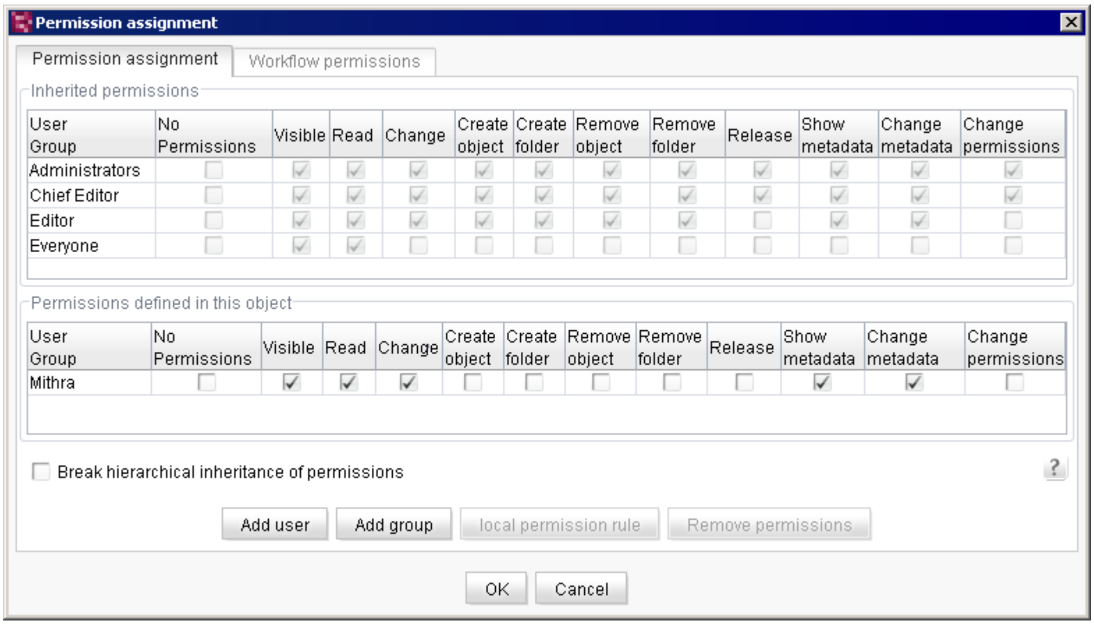
Permissions defined in this object

User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release	Show metadata	Change metadata	Change permissions
Administrators	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Chief Editor	☐	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Editor	☐	☑	☑	☑	☑	☑	☑	☑	☐	☑	☑	☐
Everyone	☐	☑	☑	☑	☑	☑	☑	☑	☐	☐	☐	☐

Figure 8-6: Permission assignment dialog at root node level

The permissions for the root node can be defined here as usual.

- **Permission assignment dialog – adding group "Everyone":** If the group "Everyone" is added on a node in the permission assignment dialog by means of the button "Add group", the hierarchy of inheritance is broken. All permissions are deprived, except for the permissions for the group "Everyone" and permissions which have been already assigned for the respective node in the area "Permissions defined in this object". The following configuration



Inherited permissions

User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release	Show metadata	Change metadata	Change permissions
Administrators	□	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Chief Editor	□	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑	☑
Editor	□	☑	☑	☑	☑	☑	☑	☑	□	☑	☑	□
Everyone	□	☑	☑	□	□	□	□	□	□	□	□	□

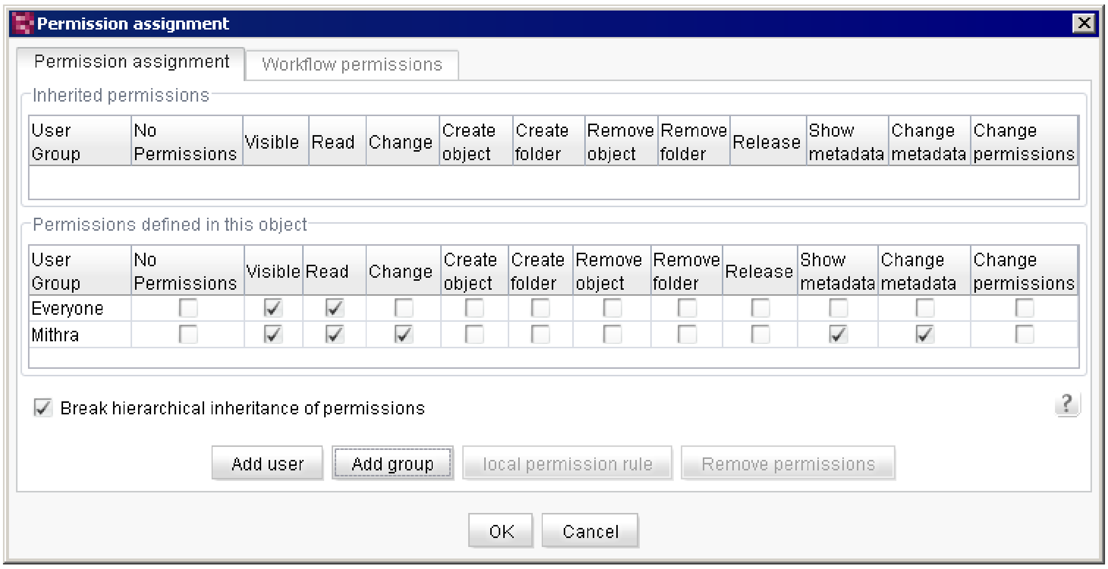
Permissions defined in this object

User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release	Show metadata	Change metadata	Change permissions
Mithra	□	☑	☑	☑	□	□	□	□	□	☑	☑	□

□ Break hierarchical inheritance of permissions

Figure 8-7: Adding group "Everyone" – Initial configuration

will become



Inherited permissions

User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release	Show metadata	Change metadata	Change permissions

Permissions defined in this object

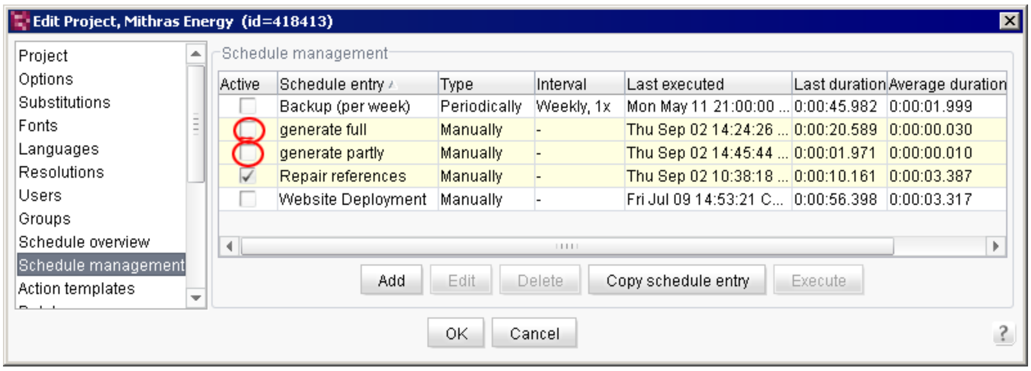
User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release	Show metadata	Change metadata	Change permissions
Everyone	□	☑	☑	□	□	□	□	□	□	□	□	□
Mithra	□	☑	☑	☑	□	□	□	□	□	☑	☑	□

☑ Break hierarchical inheritance of permissions

Figure 8-8: Group "Everyone" is added

after having added the group "Everyone".
From FirstSpirit Version 4.2R4 a corresponding notice will be displayed in this case. If it is confirmed with "Yes", the behaviour is like described above, if "No" or "Cancel" is selected, all selected groups will be added, except for "Everyone". If only "Everyone" was selected, no modification will be effectuated if you choose "No" or "Cancel".

- **Visualising objects without permissions:** If a user has got no permissions on an object, it will be hidden completely in the breadcrumb navigation of the JavaClient or *not available* is displayed. Format templates for which the user has got no permissions will not be shown in the combobox in the DOM Editor or DOM Table, as well, and thus can not be selected. Previously, ***No access*** was displayed in both cases.
- **Workflow e-mails / copy FirstSpirit address – references to data records:** If a workflow is "forwarded", a notification e-mail can be sent to users. This notification e-mail can contain a reference to the relevant nodes in JavaClient, with which the user can jump directly to the object concerned (wildcard `%FIRSTspiritURL%` or `%FIRSTspiritSOCKETURL%`, see also *FirstSpirit Manual for Developers (Part 1: Basics)*, "E-mail tab" chapter). In the case of data records, until now this reference only led to the data source, from 4.2R4 and higher it now leads to the data record itself. The "Copy FirstSpirit URL" function ("Extras" menu) now also takes into account individual data records.
- **WebEdit Help**: From 4.2R4, the WebEdit Help will no longer be HTML based but will be provided as a PDF instead. Depending on the context in which Help is opened, the relevant chapter from the latest version of *FirstSpirit Manual for Editors (WebEdit)* is displayed. The language in which the PDF document is displayed depends on the client language set via the FirstSpirit start page.
- **Updating modules**: The button "update" in the application for Server configuration (area "Modules") has been removed in 4.2R4. Updating can now be effected by using the button "Install".
- **Schedule management:** For additional protection against inadvertent starting of system schedules from JavaClient (e.g. "Project" / "Generate project" or "Generate partial project" menu) the corresponding menu entries can now be deactivated in JavaClient. This is done by deactivating the "Active" option in the project properties (application for "Server and Project Configuration"). Because, unlike previously, this option can now also be deactivated for system schedules (yellow background):

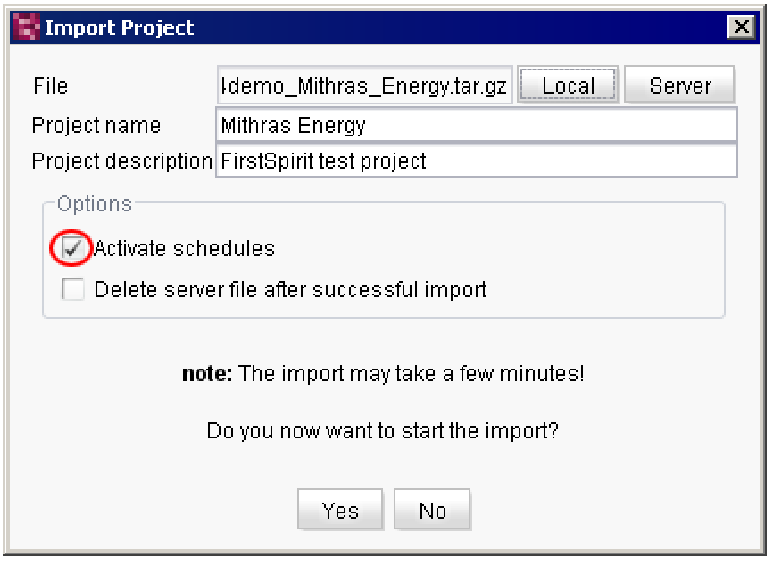


Edit Project, Mithras Energy (id=418413)

Active	Schedule entry	Type	Interval	Last executed	Last duration	Average duration
□	Backup (per week)	Periodically	Weekly, 1x	Mon May 11 21:00:00 ...	0:00:45.982	0:00:01.999
□	generate full	Manually	-	Thu Sep 02 14:24:26 ...	0:00:20.589	0:00:00.030
□	generate partly	Manually	-	Thu Sep 02 14:45:44 ...	0:00:01.971	0:00:00.010
☑	Repair references	Manually	-	Thu Sep 02 10:38:18 ...	0:00:10.161	0:00:03.387
□	Website Deployment	Manually	-	Fri Jul 09 14:53:21 C...	0:00:56.398	0:00:03.317

Figure 8-9: Project properties – Schedule management

The schedules can then not be started in JavaClient, as the menu entries are greyed out; they are also greyed out for project administrators and the server administrator(s). Regardless of the selection for "Interactive execution", schedules can be started by server and project administrators in JavaClient.
In new schedules created the "Active" option is activated as a default, as well as the system schedules in new projects created, so that as a default they can be started by the server and project administrators in JavaClient.
If a project is imported (used for "Server and project configuration" / "Project" / "Import"), until now the "Active" option of the system schedules was automatically active after the import, from 4.2R4 and higher the "Activate schedules" option must be activated in the import dialog so that the "Active" option is active:



Import Project

File: Idemo_Mithras_Energy.tar.gz

Project name: Mithras Energy

Project description: FirstSpirit test project

Options

☑ Activate schedules

☐ Delete server file after successful import

note: The import may take a few minutes!

Do you now want to start the import?

Figure 8-10: Project import

- **Rollout directory for client applications:** If you do not specify any path to a directory in which client applications (e.g. the Mozilla engine for use of the Integrated preview) are to be stored on the workstation computer by means of the parameter `CLIENT_HOME_DIR` in the configuration file `fs-server.conf`, client applications are rolled out by default from 4.2R4 into an individual directory, depending on the used FirstSpirit Version. The name of the directory contains the number of the respective FirstSpirit major, minor and release version, e.g. `\.firstspirit_4.2R4`. The directory used until 4.2R2 `\.firstspirit` will not be deleted by the system, but it must be deleted manually (e.g. to free memory).
- **Reduced default value for `CACHE_PERCENT`:** The default value for the percentage size of the cache (`CACHE_PERCENT` parameter in the `fs_server.conf` file) has been reduced from 40 to 25. It is to be expected that

this value will have to be individually adjusted.

- **LDAP configuration parameter LDAP.IMPORT_USER.LOGIN_ATTRIBUTE:** If an invalid value is given for this parameter (e.g. LDAP attribute returns an "empty" value, invalid LDAP attribute, etc.), this is logged in the server log file:

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

8.3 Expiring functions in FirstSpirit Version 4.2R4

The following functions are removed with FirstSpirit Version 4.2R4:

- **NTLM login and Windows 7:** If Microsoft Windows 7 or Vista is used, the NTLM login module, via which the NTLM authentication is made at the FirstSpirit Server, can only be used if the Windows 7 security settings are lowered. Login via Kerberos ticket (integrated Windows login) is possible in combination with Mozilla Firefox and Microsoft Internet Explorer from V4.2R2 and in this context is the preferred variant. For notes about the login via Kerberos ticket in combination with the FirstSpirit module "Personalisation" see Chapter 7.2.1page 156.
- **Release data records:** The "Release all rows" function in the Content Store in the JavaClient, in the context menu under "Extras" on data sources, has been renamed with V4.2R4 to "Release displayed lines" (see also Chapter 4.2.4 page 49).
- **Diff Program:** The option of selecting a program to compare text files in JavaClient under "Global settings" / "User settings" / "Browser" tab / "Diff Program" has been removed; the field is now no longer available.
- **Translation help / version comparison:** The "Edit mode" combobox in the translation help / version comparison dialog in JavaClient has been deleted, so that it is now only possible to open the translation help or the version comparison on a node. The "Translation help" menu item in the "Extras" menu can now only be opened on pages and sections in the Page Store or on data records of the Content Store, the menu item is greyed out on other nodes. The "Post-translate" mode has been removed, as until now it involved the same function as "Translate".
- **WebEdit themes:** The "WEBedit Theme" combobox in the Project properties (application for Server and Project Configuration / "Project" / "Webedit settings" area) is no longer available from 4.2R4.

8.4 Announcements for FirstSpirit Version 5.0

While, as a minor version, the R4 line only contains a few changes to central parts of the software, which requires manual adjustments in projects, the coming FirstSpirit Version 5.0 is a major version, in which extensive enhancements and optimisations with regard to the software interface (GUI) are planned, especially for the WebClient, but also with regard to the server backend. Therefore, project migration work will be necessary on updating to Version 5.0. Changes to the project templates will be necessary, especially if Easy Edit and Content Highlighting are used.

In addition, several internal data formats will change and will be incompatible. There will be an upgrade path from the last release version of 4.2 to Version 5, though this cannot be guaranteed for all export data formats. It is therefore possible that, e.g. a translation export made with Version 4.2, can no longer be imported following project migration to Version 5.0 (this applies by analogy to partial exports and packages). On the other hand, it will also be possible to read in a complete project export from Version 4.2 into Version 5.0, to enable project-wise migration to Version 5.0 if necessary. Due to these data format changes, automatic conversions will be performed which, depending on the project size, may have a significant runtime.

With Version 5, the last release version of 4.2 changes to long-term servicing, long-term servicing for Version 3.1 expires.

Planned changes in detail:

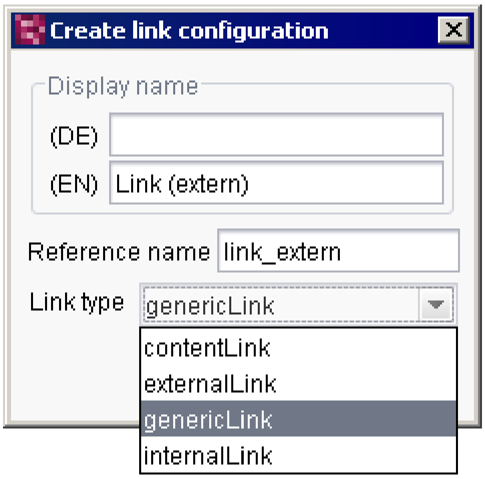
- **WebEdit:** In Version 5.0 there will be a completely new WebClient, which is visually and functionally significantly different from the previous versions. The maxim is to enable the easiest possible use by the editor. This will also affect, e.g. the design of the input components. The objective is not to display the input components familiar to date with complete functional equivalence in WebEdit 5.0, but instead to implement a variant which can be used as intuitively as possible, and therefore may have reduced functions. For example, for the FS_LIST variant in WebClient, it is planned that it will not necessarily provide all functions as in JavaClient (see Chapter 4.1.1 page 25 and Chapter 5.3 page 86), but will primarily provide those required and manageable for occasional users, in order to reduce the complexity and to ensure greater usability and simplicity.
  In addition, the horizontal Quick-Edit bar used to date will no longer be supported and will be replaced by Easy Edit as well as the new vertical Quick-Edit bar introduced in 4.2R4 (see Chapter 4.3.2 page 56).
- **Replacement of CMS_INPUT_ by FS_ input components:** It is planned to release the new input components with name prefix "FS_" introduced since FirstSpirit Version 4.2 with FirstSpirit Version 5.0. In this case, the following input components will be removed in Version 5.0:
  - CMS_INPUT_FILE
  - CMS_INPUT_PICTURE
  - CMS_INPUT_PAGEREF
  - CMS_INPUT_OBJECTCHOOSER
  - CMS_INPUT_CONTENTAREALIST
  - CMS_INPUT_CONTENTLIST
  - CMS_INPUT_LINKLIST

    - CMS_INPUT_SECTIONLIST
    - CMS_INPUT_TABLIST

  These will be replaced by the input components FS_REFERENCE, FS_DATASET and FS_LIST (see also Chapter 5.1 page 74 and Chapter 5.3 page 86).

- **CMS_INCLUDE_CONTENT:** The data element CMS_INCLUDE_CONTENT will no more be supported in FirstSpirit version 5.0. CMS_INCLUDE_CONTENT will be replaced by the enhanced data element CMS_INCLUDE_OPTIONS (see also *FirstSpirit Release Notes Version 4.2* and Chapter 5.5 page 99).

- **Links:** With FirstSpirit Version 4.2, the configuration options for links were considerably enhanced by the introduction of generic link editors (link type "genericLink").



(DE)
(EN) Link (extern)
Reference name link_extern
Link type genericLink

Figure 8-11: Link types in FirstSpirit Version 4.2

The link types "contentLink" (data link), "externalLink" and "internalLink" will no longer be available from FirstSpirit Version 5.0. For this reason, use of generic link editors is recommended. To make the changeover to generic link editors easier, FirstSpirit provides a conversion function ("Convert link template") in the context menu of the JavaClient under the sub-item "Extras").

- **FirstSpirit API:** All functions of the FirstSpirit **Access API**, the discontinuation of which was announced in FirstSpirit Version 4.2R4 or before then ("deprecations", see API documentation) will be removed with FirstSpirit 5.0. In addition, there will be a revised Version 2.0 of the **Application-API** satisfying the long-term stability criteria, which from Version 5.0 and higher will be serviced and further developed according to the familiar strategies of maintaining stable APIs in FirstSpirit. It can be incompatible with the Application-API in 4.2R4 (see also Chapter 5.13 page 107).
- **Modules:** With FirstSpirit Version 5.0, all modules must be updated. For module developers, this means recompilation of their modules against the AP valid for

FirstSpirit Version 5.0.

- **System requirements:** The Oracle (Sun) JDK 7 will be supported from FirstSpirit Version 5.0 and higher.