FirstSpirit™
Unlock Your Content

FirstSpirit™ Release Notes
FirstSpirit™ Version 5.2

Version	**1.3**
Status	**RELEASED**
Date	**2015-08-18**
Department	FS-Core
Copyright	2015 e-Spirit AG
File name	RELN_EN_FirstSpirit_Releasenotes

e-Spirit AG

Stockholmer Allee 24
44269 Dortmund | Germany

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

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

e-Spirit

Table of content

1 Introduction

This document describes the newly implemented functions in FirstSpirit V5.2. As a prerequisite, the reader must be familiar with FirstSpirit™ and must have sufficient technical background knowledge. In particular, in-depth knowledge of the relevant fields (template development, administration) is required to understand chapters 5.1 to 8.

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

1.1 "Helpdesk" has been renamed "Technical Support"

As part of the continuous improvement of our services, the familiar Helpdesk feature is being reorganized. For instance, we are changing the name from "Helpdesk" to "Technical Support" In light of this, we will be gradually adapting our documentation and other systems to reflect the new name. During the transition period, please be aware that "Helpdesk" and "Technical Support" are to be treated synonymously.

2 System requirements

This chapter summarizes the key changes in system requirements for FirstSpirit 5.2.

 *For more detailed information, see the FirstSpirit Technical Data Sheet, version 5.2.*

2.1 Operating systems

In relation to **FirstSpirit Server**, the reference version for Microsoft Windows Server has been changed from 2008 R2 to 2012 in FirstSpirit version 5.2. Only passive support is now provided for Microsoft Windows Server 2008 R2. With the release of FirstSpirit version 5.2, Red Hat Enterprise Linux 7 is now also supported. As a result, Red Hat Enterprise Linux 5 will no longer be maintained.

In relation to **FirstSpirit SiteArchitect and ServerManager**, the reference version for Mac OS has been changed from 10.7 to 10.10 in FirstSpirit version 5.2. With the release of FirstSpirit version 5.2, Microsoft Windows version 8 is now also supported. Only passive support is now provided for Ubuntu 12.04 LTS. Mac OS X 10.5 through Mac OS X 10.8 will no longer be maintained.

In FirstSpirit version 5.2, improvements have been introduced for users running Mac OS X to help them copy external content in SiteArchitect.

2.2 Java environment

FirstSpirit version 5.2 and higher supports Oracle Java 8 (JDK 8), but Java 6 will no longer be maintained.

For FirstSpirit version 5.2, the reference versions for **FirstSpirit Server** and **FirstSpirit SiteArchitect and ServerManager** are now 7u79 / 7u80 and 8u31.

2.3 Web and servlet engine/application servers

Apache Tomcat version 8 is now supported, although there are some restrictions: When the application server starts up, Apache Tomcat 8 scans all the JAR files of web applications (fs5root, fs5webedit, etc.) by default to see if it can find certain classes and TLDs. The new parsing behavior leads to significantly longer start-up times (compared with Tomcat 7), but these can be reduced by adjusting the JarScanFilter configuration. To improve the start-up times even further, you can also configure Tomcat so that the web applications start up in parallel[1].

However, please note that Apache Tomcat 6 will no longer be maintained.

The use of Apache Tomcat 7 under Oracle Java 8 (JDK 1.8) requires Apache Tomcat version 7.0.57 or higher. Older versions of Apache Tomcat do not offer sufficient support for Java 8.

Up until now, a servlet engine implementing version 2.4 of the **servlet API** was required to use the FirstSpirit web applications. As of FirstSpirit version 5.2, version 3.0 of the servlet API is required instead. Any existing FirstSpirit web applications that you have created yourself must be adapted accordingly. The web applications that e-Spirit supplies together with the FirstSpirit modules have already been adapted. As a general rule, they are now based on version 3.0, although certain modules may deviate from this principle. For more information, see also the *FirstSpirit Manual for Administrators*, chapter "Web applications".
Version 3.0 of the servlet API is supported by **WebSphere** application servers version 8.0 and higher (see also https://en.wikipedia.org/wiki/IBM_WebSphere_Application_Server#Version_history). Up until now, FirstSpirit has passively supported WebSphere 7, but this support is to be withdrawn in light of the above; instead, FirstSpirit version 5.2 will start offering passive support for versions of WebSphere that are higher than 8.5.

[1] See http://wiki.apache.org/tomcat/HowTo/FasterStartUp.

2.4 Databases

With the release of FirstSpirit version 5.2, **MySQL** version 5.6 is now also supported. However, please note that MySQL 5.1 will no longer be maintained.

The reference versions for **Microsoft SQL Server** are now Microsoft SQL Server 2012 and Microsoft SQL Server 2014. Only passive support is now provided for Microsoft SQL Server 2008.

With the release of FirstSpirit version 5.2, **Oracle** version 12c and 11gR2 are now also supported. As a result, Oracle 10.x will no longer be maintained.

The reference version for **IBM DB2** is now IBM DB2 10.5. There are plans to introduce support for IBM DB2 11 in future versions of FirstSpirit.

The reference version for **PostgreSQL** is now PostgreSQL 9.3. Only passive support is now provided for PostgreSQL 8.4.

Upgrades from PostgreSQL version 8 to version 9 are particularly prone to problems as far as the use of FirstSpirit is concerned. In such cases, the version of the database may not (may no longer) be compatible with the version of the JDBC driver used with FirstSpirit. Therefore, if you do encounter any problems with the JDBC driver, you should update it to a version that is compatible with the version of the database (see also the FirstSpirit Manual for Administrators, chapter "Storing the JDBC driver files" and onward).

2.5 Web browsers (for using FirstSpirit ContentCreator)

The reference version for **Microsoft Internet Explorer** has been increased from 9 to 10. In addition, **Google Chrome** 43 has now also been added as a reference version.

On top of that, version 11 of **Microsoft Internet Explorer** is now also supported. As a result, versions 9 and 8 will no longer be maintained.

In addition, FirstSpirit now offers passive support for **Apple Safari**.

Please also be aware that Mozilla Firefox 3.5, 3.6 and 4 will no longer be maintained.

2.6 Internal software

The following internal software has also been updated in FirstSpirit 5.2:

- **Apache Derby** (integrated database for test purposes):
  Update from version 10.8.2.2 to version 10.11.1.1
- **Apache FOP** ("Formatting Objects Processor", enables you to output FirstSpirit content in PDF format, for example):
  Update from version 1.0 to version 1.1

Change in behavior for fonts that utilize "Private Use Areas" (PUAs): As per the FOP mailing list, the way certain fonts are handled has changed with the release of version 1.1[2]. This means, for example, that you must now use the PUA code instead of the value from the Character Map when using the Wingdings font. To display character 0x78, you must now enter "ampersand + #xF0 + 78" in FOP.
This applies to all fonts that utilize PUAs (for more information, see http://de.wikipedia.org/wiki/Private_Use_Area).

- **Eclipse Jetty** (integrated web server):
  Update from version 8.1.3 to version 9.2.9

If you are using Eclipse Jetty as a web server (parameter in configuration file fs-server.conf): "INTERNAL_SERVLET_ENGINE=1"), you will have to adapt file fs-webapp.xml manually to make it compatible with existing installations due to an incompatible change that occurs during an in-place upgrade to FirstSpirit version 5.2. This file contains the Jetty configuration settings and can be found inside directory "~/conf/". Unless you perform this step, Jetty will no longer start. For more information, see the FirstSpirit installation instructions, chapter "If using the integrated web server Jetty".

- **Apache Lucene** (program library for full-text search):
  Update from version 3.6.0 to version 4.8.1

[2] See also http://mail-archives.apache.org/mod_mbox/xmlgraphics-fop-dev/201209.mbox/%3CCACQ=j+d4xfR6qLr1LfyqiyNzMYR8ANwQtUsFhPm3_2MbnN3ytA@mail.gmail.com%3E

- **JxBrowser** (integrated preview):
  Update to version 4.9. As a result, for example, PDFs (e.g., Help PDFs) can now be displayed in the integrated preview if you select Google Chrome as the browser engine in SiteArchitect (menu: "View / Browser Engine / Google Chrome").
- **SLF4J** (Simple Logging Facade for Java):
  The library SLF4J has been updated from version 1.6.4 to version 1.7.12.

In addition, the Java wrapper version has been updated to 3.5.26.

3 Switching from older FirstSpirit versions

For information on new installations and upgrading to FirstSpirit version 5.2, see the *FirstSpirit Installation Instructions* for version 5.2.

The software only supports **updates** that take you from the latest release version of FirstSpirit 5.1 to version 5.2. In principle, it is still possible to perform an update from other FirstSpirit versions, but this process is not supported officially.

Similarly, it is still possible – in principle – to **downgrade** from version 5.2 to the last release version of FirstSpirit 5.1, although this process is not directly supported either. In such cases, please contact e-Spirit direct.

If you are using **Eclipse Jetty** as a web server (parameter in configuration file fs-server.conf): "INTERNAL_SERVLET_ENGINE=1"), you will have to adapt file fs-webapp.xml manually to make it compatible with existing installations due to an incompatible change that occurs during an in-place upgrade to FirstSpirit version 5.2. This file contains the Jetty configuration settings and can be found inside directory "~/conf/". Unless you perform this step, Jetty will no longer start. For more information, see the FirstSpirit installation instructions, chapter "If using the integrated web server Jetty".

In addition, whenever you perform a FirstSpirit server update, we recommend that you regenerate any **modules** that you have created yourself under the latest FirstSpirit version. All installed modules provided by e-Spirit and all web applications should be updated as well!

4 New/changed functions for all user groups

4.1 Online documentation

The online documentation for FirstSpirit offers extensive information about template development as well as about working with FirstSpirit in general. The documentation URLs have been updated for FirstSpirit version 5.2. Bookmarks to the FirstSpirit online documentation should, therefore, be adapted accordingly.

4.2 Start page and favicons

The button for starting ContentCreator on the FirstSpirit start page has been slightly redesigned. In addition, the start page now also includes an icon for using the FirstSpirit Launcher (see chapter 7.4, page 126 and the FirstSpirit installation instructions). The buttons for starting FirstSpirit SiteArchitect and ServerManager can be provided with a Launcher icon:

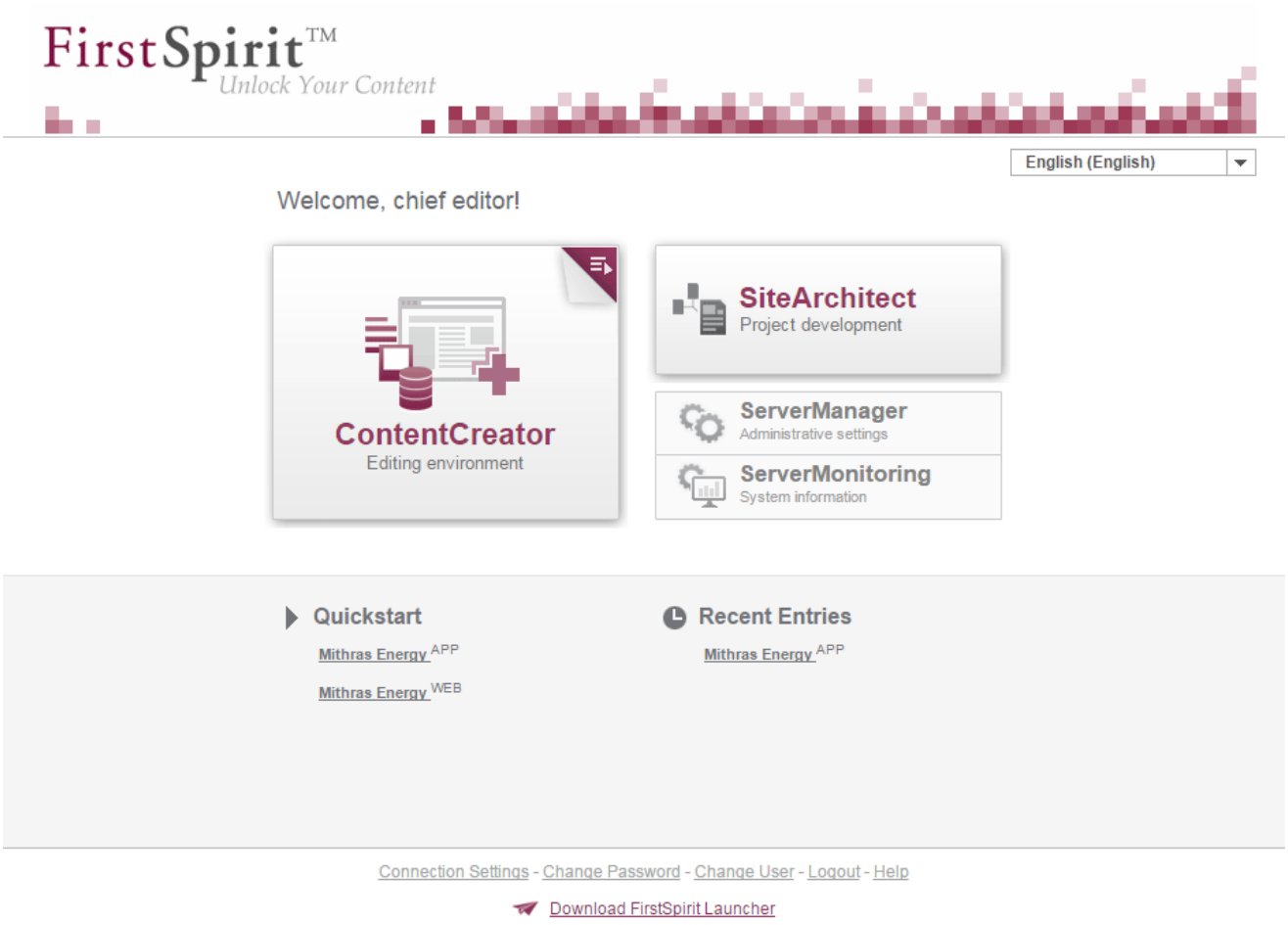


Figure 4-1: The FirstSpirit start page

FirstSpirit version 5.2 also introduces new favicons for the various FirstSpirit services:

- Start page and preview:
- Web applications:
- ServerMonitoring:
- ContentCreator:
- Online documentation:

4.3 New input components

As part of FirstSpirit version 5.2, two new input components have been introduced which support functions for bundling content: FS_CATALOG and FS_INDEX. FS_CATALOG and FS_INDEX have been introduced to simplify configuration and improve usability, particularly as far as nested components are concerned. The operating concept has been inspired by the design and functionality of the data store, which means that editors always edit internal elements in the (central) workspace and can see an overview of the existing entries on the left-hand side of the screen.

- FS_CATALOG:

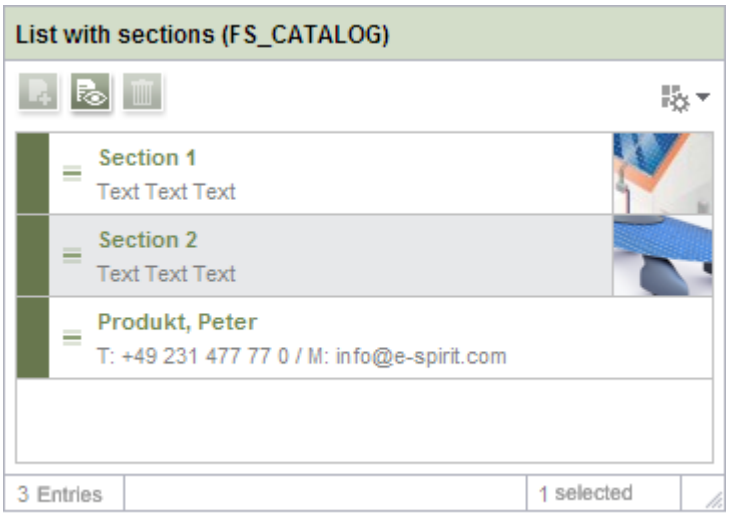


Figure 4-2: FS_CATALOG in SiteArchitect

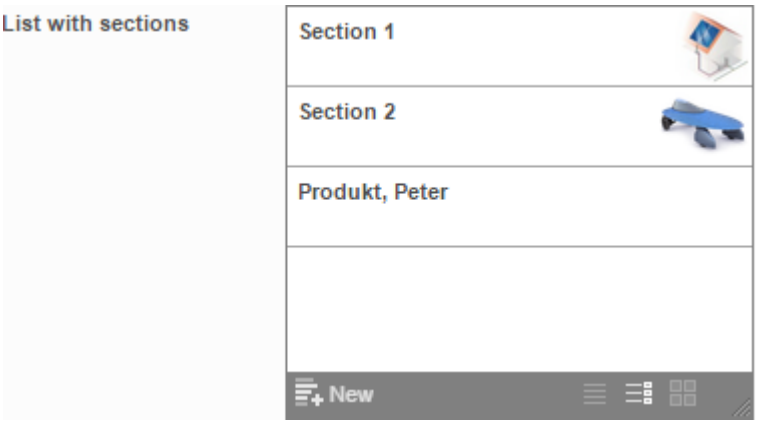


Figure 4-3: FS_CATALOG in ContentCreator

The FS_CATALOG input component makes it possible to create lists of sections or links so that editors can maintain multiple sections/links directly in one input element without having to switch to a different input component each time they want to access another section, etc. FS_CATALOG functions in a very similar way to the "Inline" type of the FS_LIST input component.

- FS_INDEX:

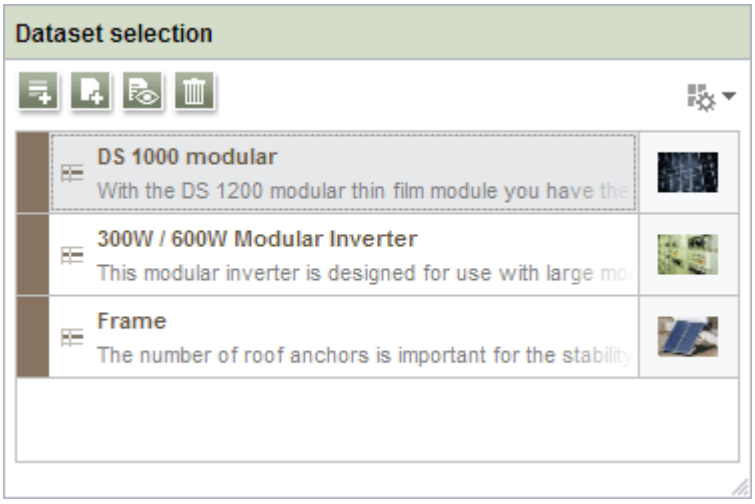


Figure 4-4: FS_INDEX in SiteArchitect

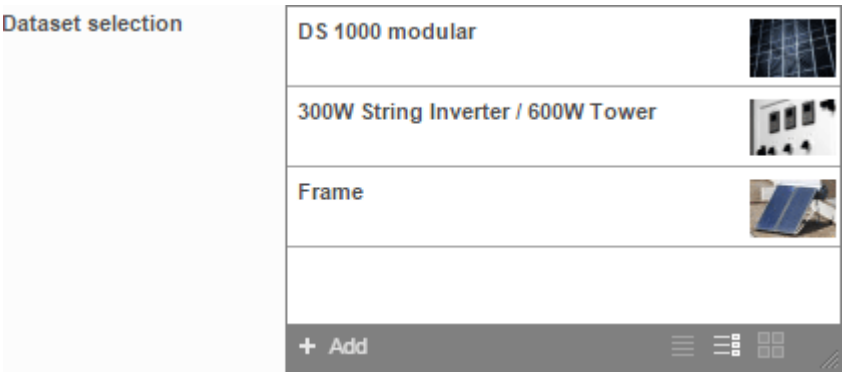
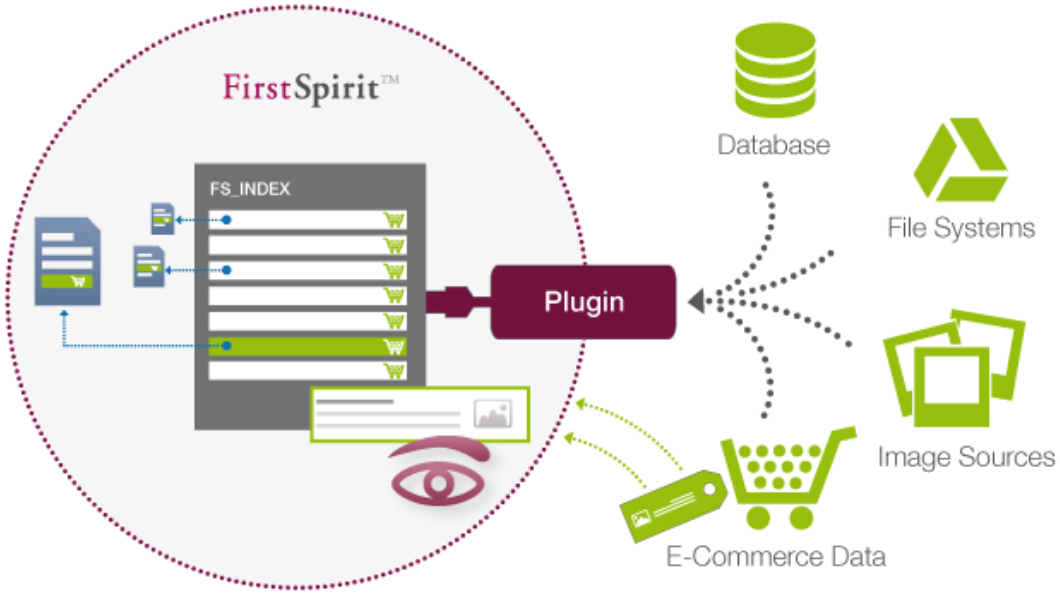


Figure 4-5: FS_INDEX in ContentCreator

The FS_INDEX input component allows you to link external components or modules so that they can supply the input component with data. The data can then be selected (i.e., referenced) via FS_INDEX. The function and appearance are highly dependent on how the module is individually implemented.



These new input components are accompanied by new data types:

- Catalog (returned by FS_CATALOG)
- Card (part of the catalog data type)
- Index (returned by FS_INDEX)
- Record (part of the index data type)

For detailed information, see also

- FirstSpirit Online Documentation, "Template development / Forms / Input components / CATALOG"
- FirstSpirit Online Documentation, "Template development / Forms / Input components / INDEX"
- FirstSpirit SiteArchitect documentation, "Catalog (FS_CATALOG)" chapter
- FirstSpirit SiteArchitect documentation, "Index (FS_INDEX)" chapter
- FirstSpirit ContentCreator documentation, "Editing a preview page / Input elements / Catalog"
- FirstSpirit ContentCreator documentation, "Editing a preview page / Input elements / Index"

For more information about FS_CATALOG and FS_INDEX, see also chapter 6.1, page 64.

The current FS_LIST implementation is being maintained for the time being for reasons of compatibility. However, its discontinuation (deprecation) is being planned to coincide with FirstSpirit version 6.0. It will be completely removed in a later version.

4.4 Enhancements concerning bookmarks

As of FirstSpirit version 5.2, bookmarks and master copies can be made available to all users on a project-wide basis, making it easier for them to collaborate. Special bookmark groups can be created for this purpose in **SiteArchitect** (see chapter 4.4.1, page 18).

Only project administrators (including super and server administrators) are permitted to create project-wide bookmark groups and to add and remove elements of these groups. Moreover, these actions can only be performed in SiteArchitect.

4.4.1 Creating project-wide bookmark groups in SiteArchitect

To create bookmark groups in SiteArchitect, go to the "Organize" area. Click the  icon to open the "Create bookmark group" window shown below:

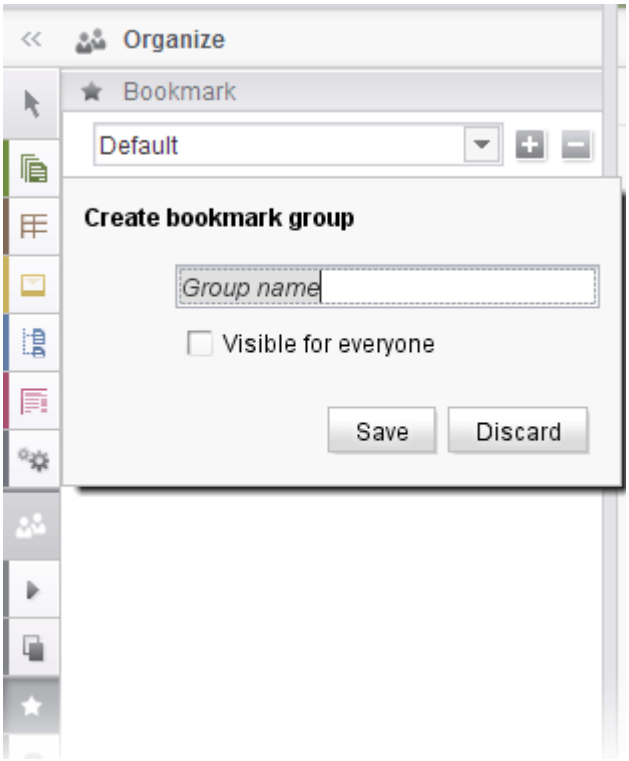


Figure 4-6: Create bookmark group

You can change the default text **Name of the group** as required.

Visible for everyone: If this option is *checked*, the group will be visible to all users of the current project, along with all the elements that are added to the group (however, please note that not all element types can be displayed in ContentCreator as bookmarks; for more information, see chapter 4.4.3, page 20). The suffix "(Project)" is automatically added to the name of the group.

This option is only available to project administrators.

Click "Save" to create the new group or "Discard" to close the window without saving a new group.

4.4.2 Adding bookmarks to and removing bookmarks from a project-wide group

When creating a new bookmark in SiteArchitect, project administrators have access to all the project-wide bookmark groups and can add elements to these groups. If the

elements concerned are located within groups for which the **Visible for everyone** option has been checked (see Figure 4-6), all the users of the current project will see them listed under their own bookmarks (exception: ContentCreator; see chapter 4.4.3, page 20). Even the order of the bookmarks is respected.

If a bookmark is no longer required, it can be removed by clicking the × icon that appears next to it.

Bookmarks can only be added to/removed from project-wide bookmark groups by project administrators.

For more information on working with bookmarks in SiteArchitect, see also the FirstSpirit SiteArchitect documentation, chapters "New bookmark" and "Bookmarks".

4.4.3 Bookmarks in ContentCreator

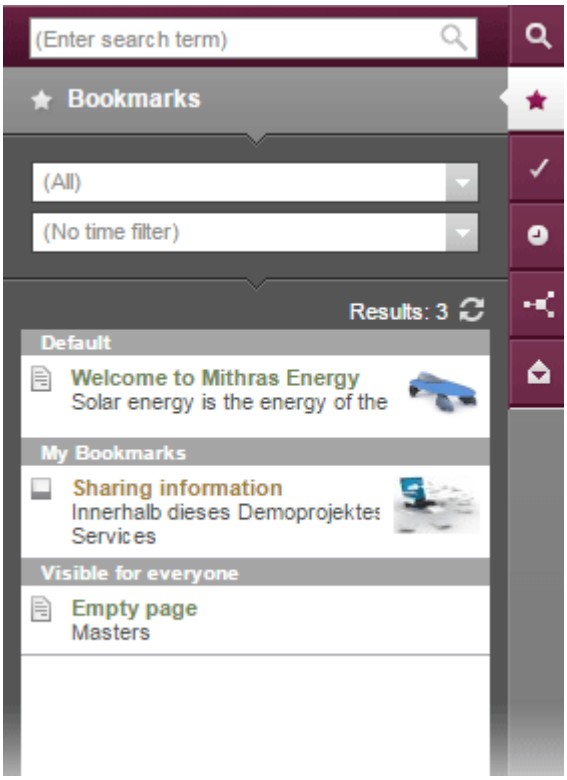


Figure 4-7: Bookmarks in ContentCreator

In ContentCreator, any bookmarks that are available on a project-wide basis are shown in the bookmark report. However, not all element types (e.g., folders and templates) can be displayed as bookmarks because of how the software has been designed.

Bookmarks can only be added to/removed from project-wide bookmark groups by project administrators.

In previous versions of FirstSpirit, it was only possible to display one bookmark group at a time in ContentCreator. By contrast, FirstSpirit version 5.2 (and higher) allows all the bookmark groups that are available for the current users to be displayed. The bookmarks can be filtered by group using the group filter (the "(All)" option in Figure 4-7).

In addition, it is now possible to display sections and section master copies in the bookmark report as well.

For more information on working with bookmarks in ContentCreator, see also the following pages of the FirstSpirit ContentCreator documentation: "Menu functions / History area / Bookmarks" and "Report area / Bookmarks".

4.5 SiteArchitect: Comparing and merging changes

With the release of FirstSpirit version 5.2, it is now possible to incorporate a program for comparing and merging files (such as WinMerge) into SiteArchitect. This allows you to track and merge changes in the version history.

4.5.1 Selecting the required program

You can incorporate the required program into SiteArchitect by selecting "User settings" under "Global Settings" and going to the "Merge" area:

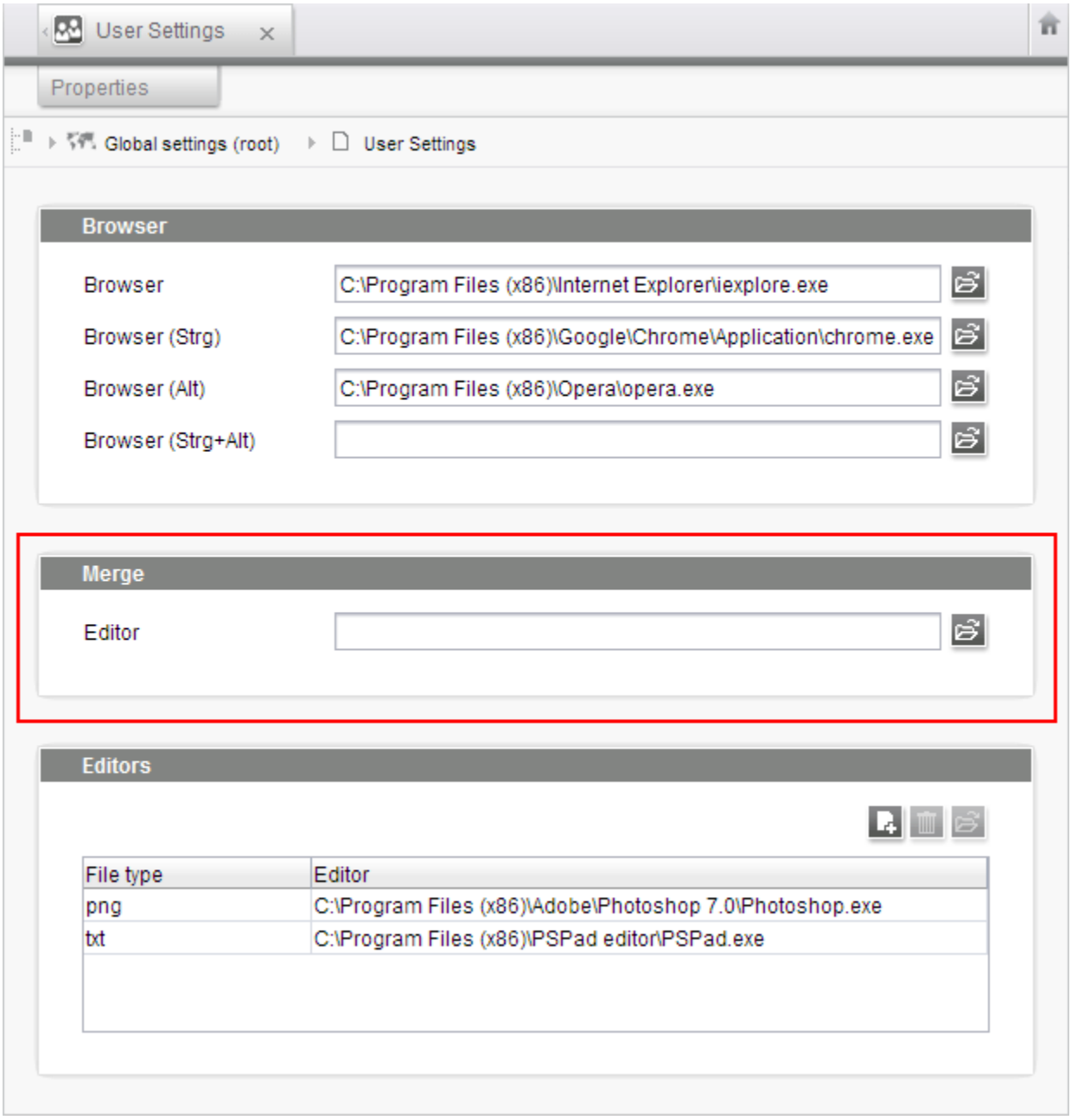


Figure 4-8: User settings – Merge program

You must select a program that can be called via the command line.

By default, the selected program is called with the following three parameters:

- %1: This is a placeholder for the version of the file that is to be opened on the **left-hand** side of the version comparison in SiteArchitect ("original").

- %2: This is a placeholder for the version of the file that is to be opened on the **right-hand** side of the version comparison in SiteArchitect ("modified").
- %3: This is a placeholder for the file where the results of the merge process are to be saved ("merge result").

You can use these parameters to control how the different versions are arranged in the selected program.

Using the call

```
"C:\Program Files (x86)\WinMerge\WinMergeU.exe" %1 %2 %3
```

in conjunction with the function would, for example, open the "WinMerge" program with the files arranged as follows: The file shown on the left-hand side of the SiteArchitect version comparison would be displayed on the left in the merge program and the file shown on the right-hand side of the SiteArchitect version comparison would be displayed on the right in the merge program.

These parameters must be present, otherwise an error message will be output later on when the program is called (see chapter 4.5.2, page 23). Depending on the syntax of the program concerned, it may be necessary to adjust these parameters.

4.5.2 Comparing and merging changes in version comparisons

The program that has been selected as described in chapter 4.5.1 on page 21 can be called in version comparisons (with the exception of (global) pages, (global) sections, datasets):

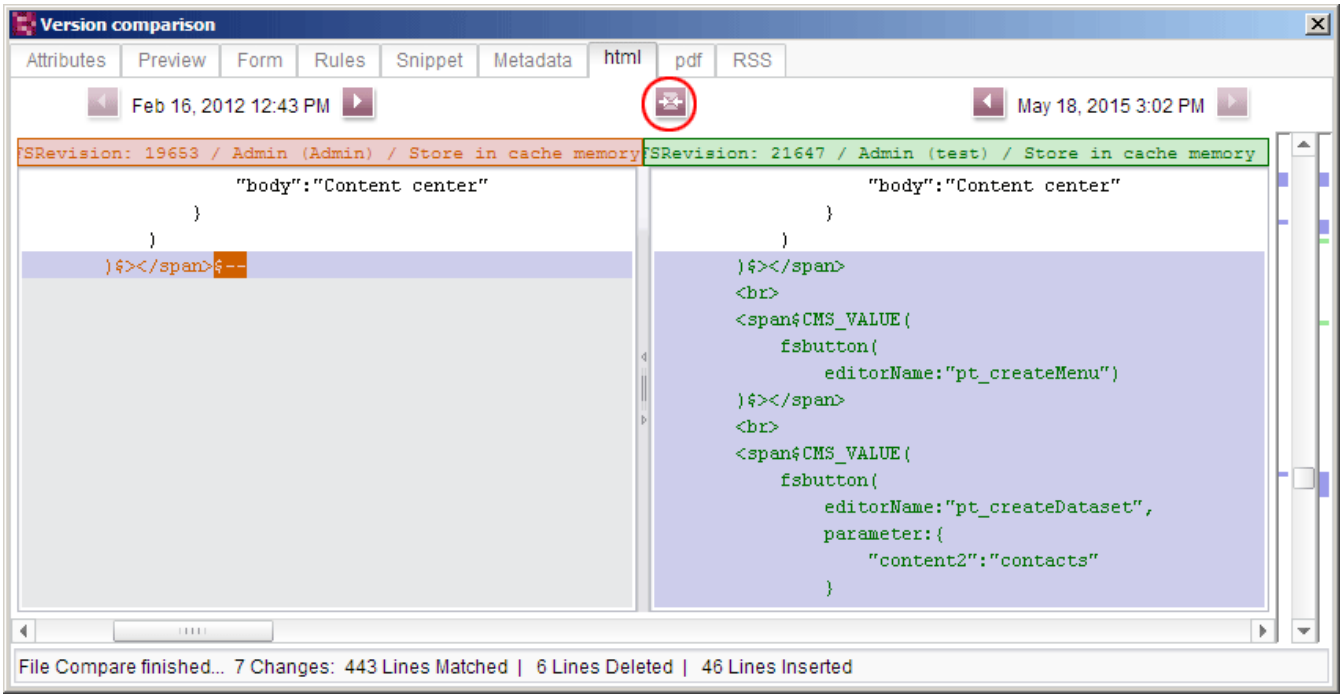


Figure 4-9: Version comparison

4.6 SiteArchitect: Searching for a CRC checksum

As of FirstSpirit version 5.2, you can now use the global search function in SiteArchitect to look for a CRC checksum that relates to a specific medium. The corresponding syntax is:

```
fs.crc = 1180948165
```

The CRC checksum of the medium being sought is entered after the equal sign.

See also FirstSpirit Developer API, `de.espirit.firstspirit.agency` package, `QueryAgent` interface.

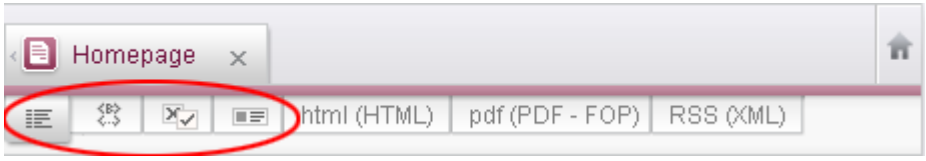
Note: If you are performing an in-place upgrade from FirstSpirit version 5.1 to 5.2 (see the *FirstSpirit Installation Instructions, version 5.2,* chapter "Via the fs-server.jar file"), the search index must be updated before you can use this function.

4.7 SiteArchitect: Display of tabs in the workspace is more compact

The display of tabs in the workspace is now more compact: Icons are used instead of labels in some stores, e.g.,

- In the site store:

- In the template store:



By deactivating the new "Compact view of tabs" menu entry in the "View" menu,

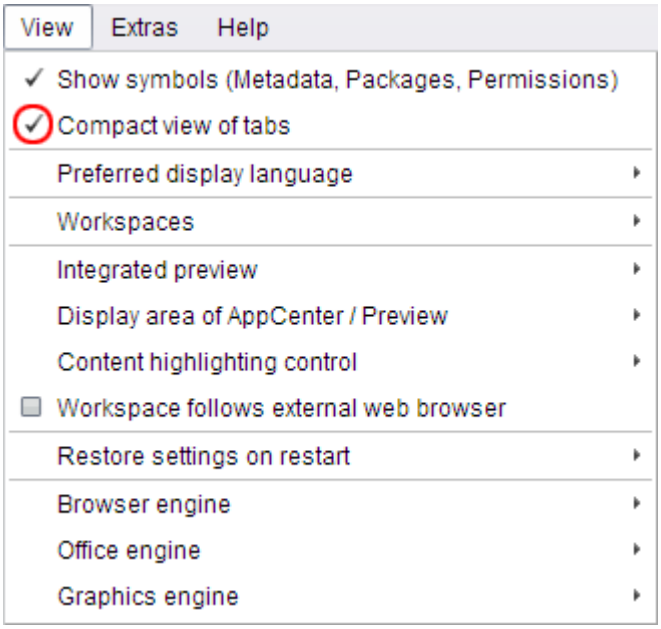


Figure 4-10: "View" / "Compact view of tabs" menu

it is possible to revert to the display settings that were used in versions of FirstSpirit prior to 5.2 ("Properties", "Menu order", "Form", "Rules", "Snippet" labels, etc.) Whatever is selected here will be saved as a server-wide setting for the current user.

5 New/changed functions for editors

5.1 New input components

As part of FirstSpirit version 5.2, two new input components have been introduced which support functions for bundling content. For more information, see chapter 4.3, page 15.

5.2 LiveEdit

As of FirstSpirit version 5.2, it is now possible to edit content directly on a page that has already been deployed (e.g., an intranet page). For this purpose, corresponding edit icons now appear so that editors can change text very quickly and more directly, or even upload images. Naturally, all the permissions of the editors concerned are evaluated during this process and, if necessary, the changes can be released by workflow.

Figure 5-1: Example LiveEdit page

 Hovering the mouse pointer over the page displays a frame and this icon to indicate where content can be edited. Clicking this icon opens an edit window containing all the input elements for the selected element (page, section, dataset). You can edit the content in this window.

For information on editing content in ContentCreator, see also the FirstSpirit ContentCreator documentation, chapter "Editing a preview page".

Depending on how the project has been configured by the project developer, another window may appear when you save the changes ("Save" button). You can then use this to start a workflow and thereby release and deploy the edited content (depending on the project configuration).

For information on working with workflows in ContentCreator, see also the following pages of the FirstSpirit ContentCreator documentation:

- *"Report area / Tasks / Workflows"*
- *"Report area / Tasks / Releases"*

5.1 New/changed functions in ContentCreator

5.1.1 Multi login via a single browser

As of FirstSpirit version 5.2, it is now possible to open multiple projects within the same instance of the browser. As a result, there are several possible scenarios:

- 1 editor, n instances of the project: The editor can open the same project on multiple browser tabs at the same time.
- 1 editor, n projects: The editor can open various different projects at the same time.

Figure 5-2: Multiple ContentCreator projects in a single browser

5.1.2 New functions in reports

5.1.2.1 Refreshing the report display

An icon for refreshing the relevant list has been added to the Report area:

Clicking this icon allows you to refresh the report entries if required.

5.1.2.2 Visualizing report entries in the preview

In ContentCreator, it is now possible to visualize report entries in the preview and filter them according to structural aspects of the project. By default, this is possible for search results ("Search" report entries, icon) and tasks ("Tasks" report entries, icon).

If you check the new "Result indicator" box in the report, the following information is shown in the preview (assuming that the project has been configured accordingly):

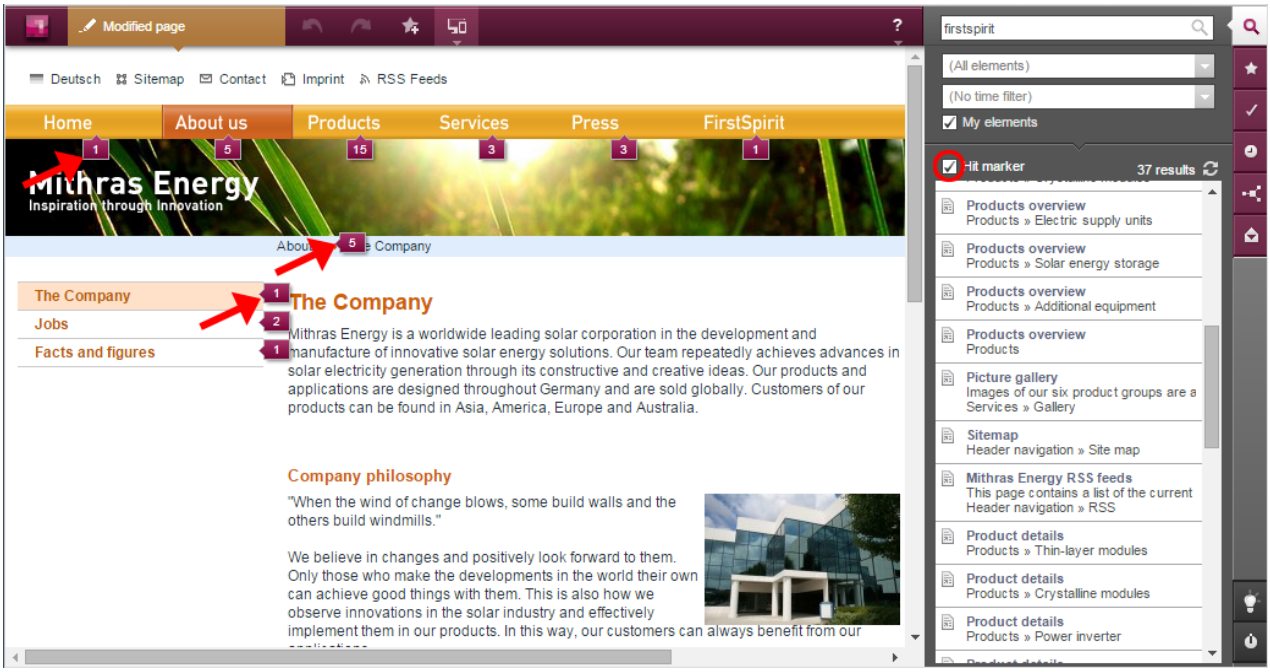


Figure 5-3: Search results in ContentCreator

5 This icon is displayed for menu items that are linked to report entries (e.g., search results). The number clearly indicates how many corresponding report entries there are for the menu item concerned.

Hovering the mouse pointer over one of these icons highlights the corresponding entries in the report by applying a background color. Clicking one of the icons hides all the entries in the report apart from those that are relevant. In other words, a filter is applied.

Clicking the "Cancel narrowing of results" link removes the filter and all the report entries are displayed again.

5.1.2.3 New report actions

Depending on what settings have been configured by the project developer, new actions may be available for project-specific reports, e.g., the ability to add new report entries.

5.1.3 Translation help

If content has already been entered in one language, the translation help function provides an easy way of copying the existing content across to other languages so that it can be used there as a basis for translation. This process places the content in the various languages side by side. The translation help must be configured accordingly in the project by the project developer. It can be opened by clicking the  icon.

 Depending on the project configuration, clicking this icon opens a dialog so that you can transfer the content from language A into corresponding input elements of language B. Multiple languages are displayed in a context menu. Click with the mouse to select the language into which the content is to be translated.

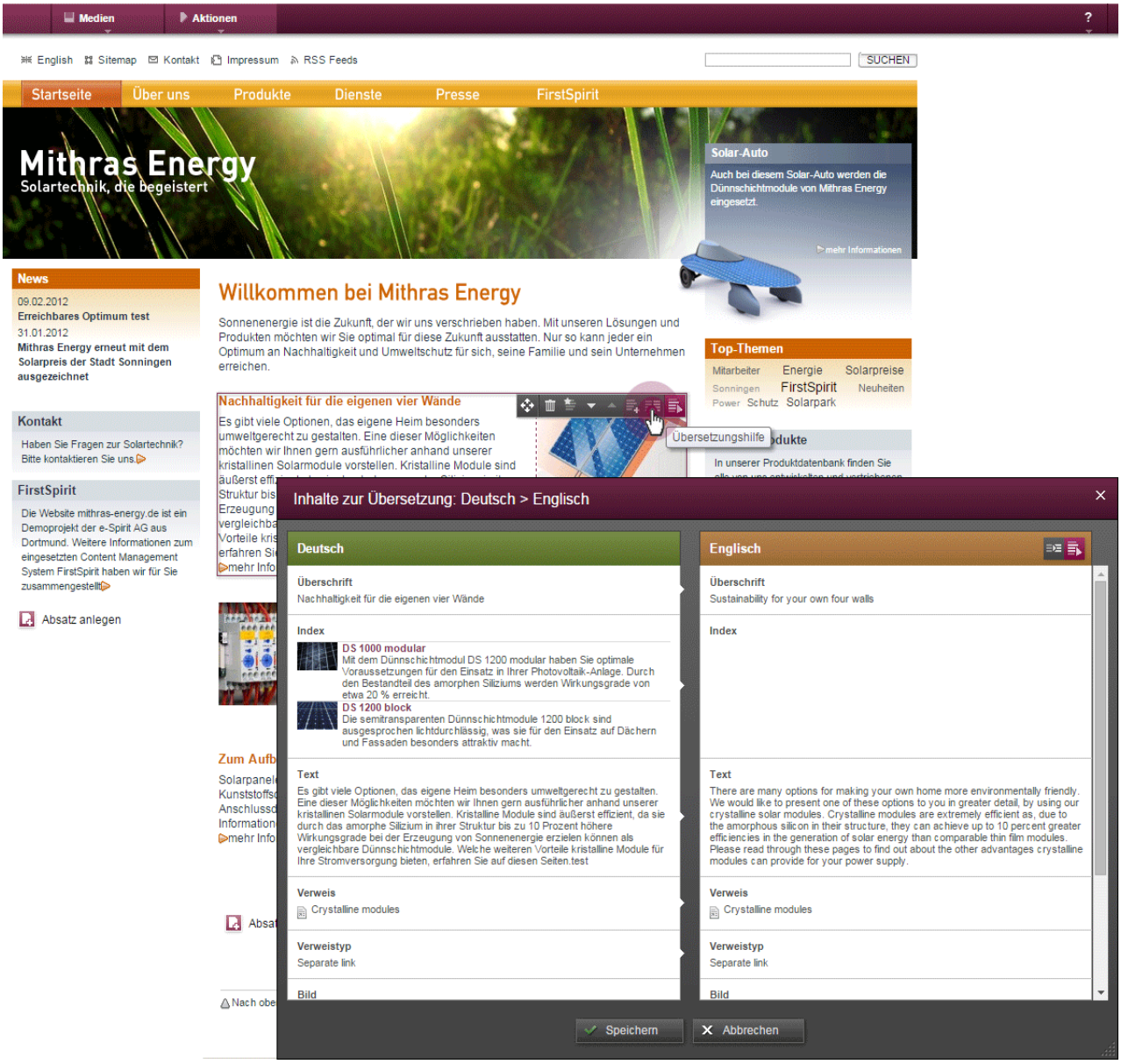


Figure 5-4: Translation help in ContentCreator (example)

Different projects and different editors may have different requirements as far as the translation process is concerned. For this reason, the software allows for the implementation of a project-specific solution. This may involve automatic (pre-)translation by an external service (e.g., Google Translate). In this case, depending on the project configuration, content that has been translated automatically will be displayed against a blue background. The blue background disappears when the content is edited by the editor.

For more information on the translation help function in ContentCreator, see also the FirstSpirit Manual for Editors (ContentCreator), chapter "Editing a preview page / Working with dialogs / Translation help".

5.1.4 Generating content with drag-and-drop

The ability to create links in ContentCreator by using drag-and-drop in the DOM editor was implemented back in FirstSpirit version 5.1. In version 5.2, this option has been extended to page content, which means that you can now also use drag-and-drop to create pages, sections, and datasets. An example of this can be found in the "Mithras Energy" sample project in the form of the "Create section" button with the  icon. Destinations such as these onto which you can drop the elements are also referred to as "drop zones". When you hold a droppable element over a drop zone, a blackberry-colored frame is applied to the zone concerned.

As a general rule, texts from external applications (word processing programs, web pages) can now also be inserted into the following input elements using drag-and-drop:

- Single-line text (CMS_INPUT_TEXT)
- Multi-line text (CMS_INPUT_TEXTAREA)

If you are using the "Google Chrome" browser and drop text from a word processing program into the "Single-line text" or "Multi-line text" input elements, the text may be removed from the external application[3]. This is not an error on the part of FirstSpirit!

For more information on the drag-and-drop function in ContentCreator, see also the FirstSpirit Manual for Editors (ContentCreator), chapter "Editing a preview page / Operating concept".

[3] See https://productforums.google.com/forum/#!topic/chrome/HoQrTZzGh-U

5.1.4.1 Pages

Depending on the project configuration, you can – for example – create new pages with the drag-and-drop function as described below:

- By dragging **pages** from the Report area to a drop zone (e.g., onto a button for creating pages)
- By dragging **pages** from the Report area to a menu item, e.g.

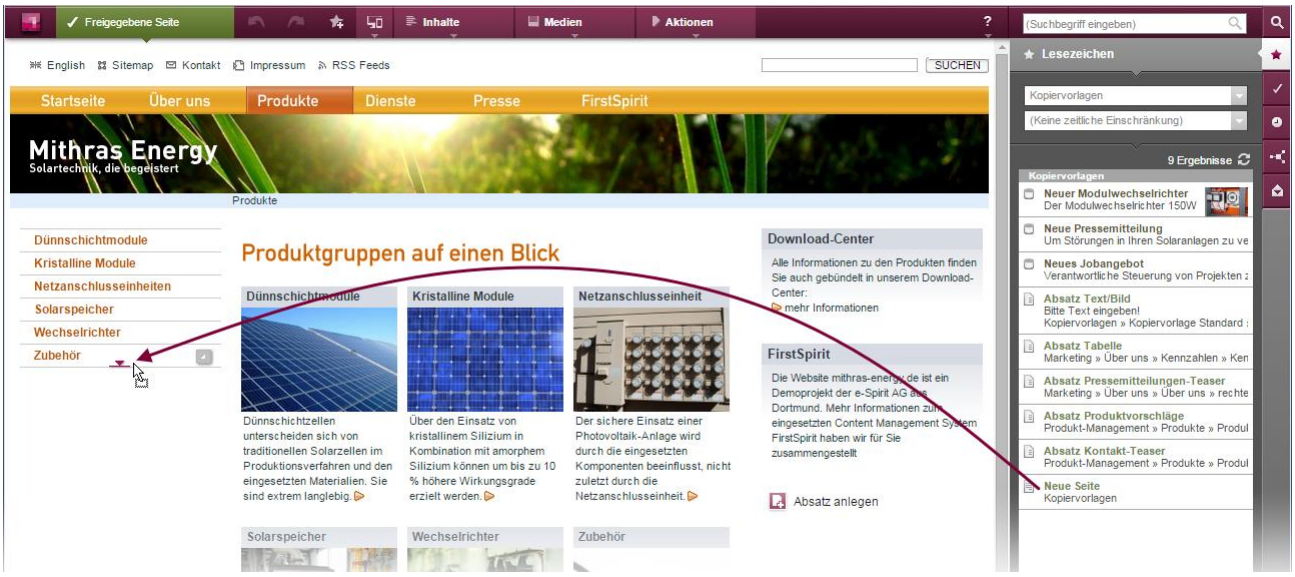


Figure 5-5: Using drag-and-drop to create pages

In the same way as when menu items are moved in the preview (see *FirstSpirit Manual for Editors (ContentCreator)*, chapter "Editing a preview page / Navigation", section "Editing navigation in the preview"), icons appear next to the navigation elements/menu entries if pages are held over them with the mouse button pressed.

As soon as the mouse button is released at the required position, the dialog for inserting a new page into the website navigation system opens:

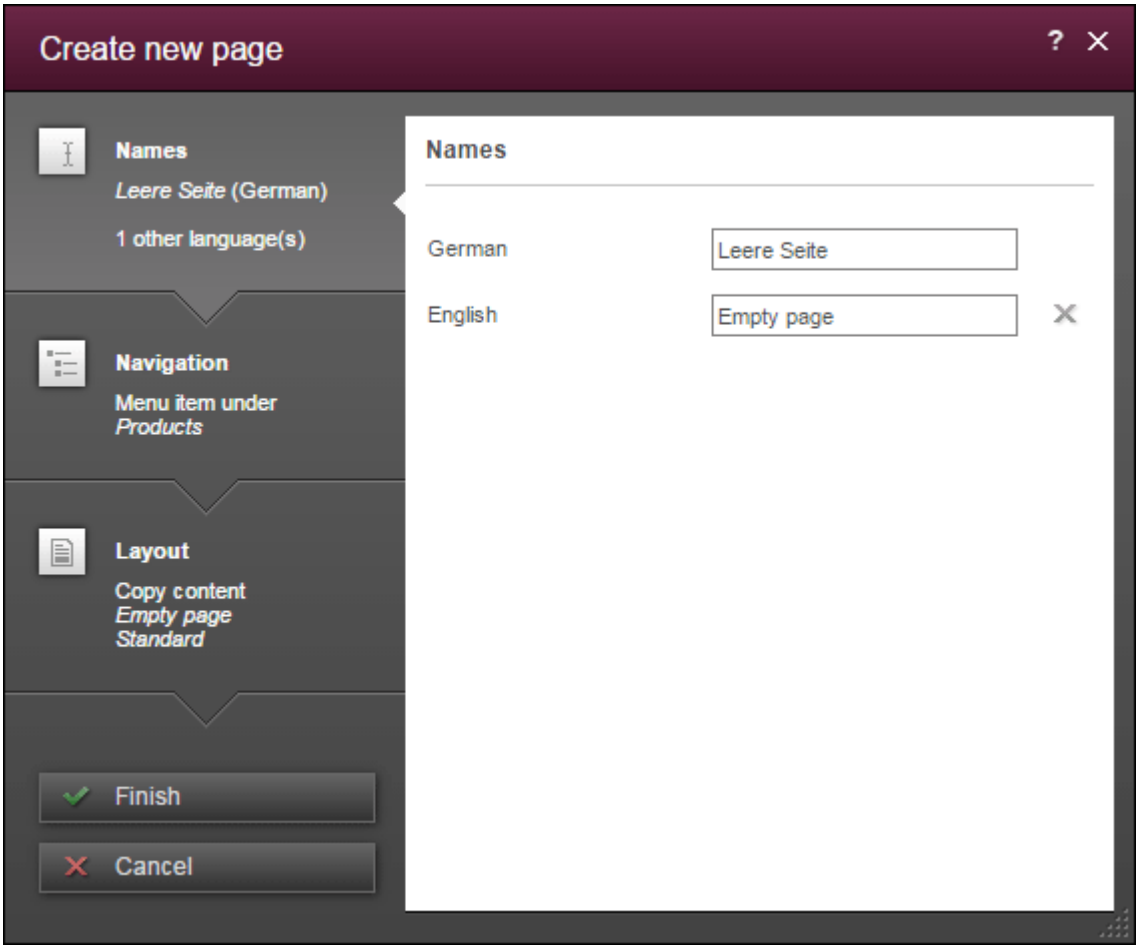


Figure 5-6: Creating a new page

When you click "Finish", the default setting leads to the creation of a new page at the required position within the navigation system that has the same name and content as the dropped page. If necessary, you can change the settings for the new page in this dialog or after it has been created.

For more information on how to edit the navigation system in ContentCreator using drag-and-drop, see also the FirstSpirit Manual for Editors (ContentCreator), chapter "Editing a preview page / Navigation"; for information on creating pages, see chapter "Menu functions / Content area / New page".

5.1.4.2 Sections

Depending on the project configuration, you can – for example – create new sections with the drag-and-drop function as described below:

- By dragging **sections/section master copies** from the Report area to a drop zone (e.g., onto the "Create section" button)
- By dragging **media** from the Report area to a drop zone (e.g., onto the "Create section" button)
- By dragging **pages** from the Report area to a drop zone (e.g., onto the "Create section" button)
- By dragging **datasets** from the Report area to a drop zone (e.g., onto the "Create section" button)
- By dragging **text** to a drop zone (e.g., onto the "Create section" button)

Color highlighting is applied to indicate where the element can be dropped.

As soon as you release the mouse button in the required area (i.e., where the section is to be created), the section is usually created right away. Section data (texts, images, etc.) can be entered subsequently ( icon) and any placeholder information stipulated by the project developer can be adapted as necessary. If a section or a section master copy has been selected, a copy of the dropped section is created in the last location. If a medium, a page, a dataset, or a text has been selected, they will automatically be included in a suitable input element (assuming that the project developer has configured the project accordingly) and will be saved at the same time.

If several section templates are available for selection (i.e., there is a choice of different layouts for the section you want to create), a selection list opens so that you can select the required template and then enter the content in the usual manner. In the same way as before, any media, datasets, pages, or text that you drop into the drop zone will automatically be included in a suitable section input element (if the project developer has configured the project accordingly), e.g.:

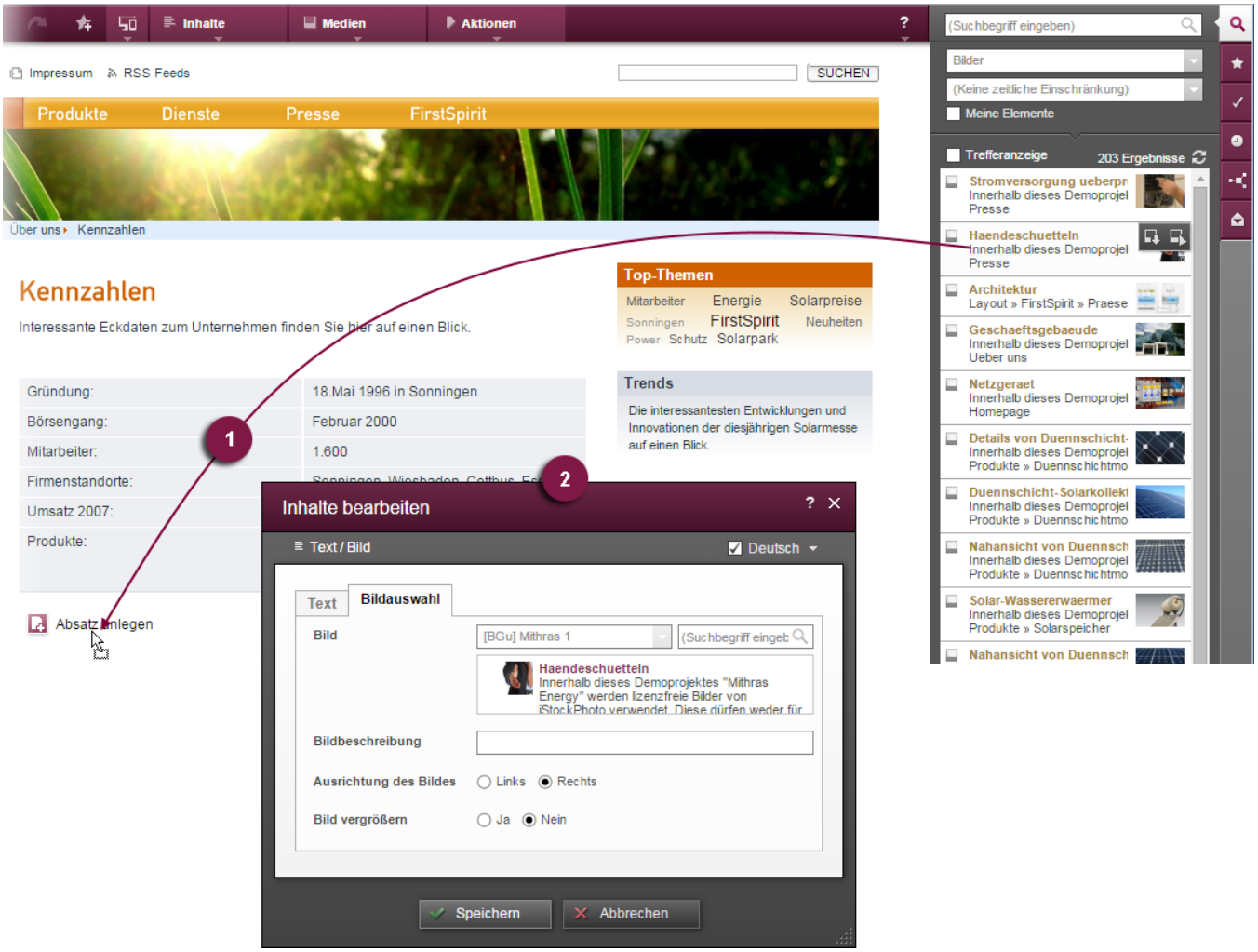


Figure 5-7: Creating a section by dragging and dropping an image

The content of the edit window can be edited as required and then saved in the usual manner.

If the new section contains mandatory fields that have to be completed by the editor, an edit window opens when the mouse button is released in the required area (i.e., where the section is to be created). This edit window contains the input elements that are available for the section concerned. If a medium, a page, a dataset, or a text has been selected, they will – once again – be automatically included in a suitable input element (assuming that the project developer has configured the project accordingly).

For more information on creating and editing sections in ContentCreator, see also the following pages in the "FirstSpirit Manual for Editors (ContentCreator)":

- *"Editing a preview page / Sections"*
- *"Report area / Bookmarks", section "Functions"*
- *"Editing a preview page / Working with dialogs / Editing window / Rule violations"*

5.1.4.3 Datasets

Depending on the project configuration, you can – for example – create new datasets with the drag-and-drop function as described below:

- By dragging **media** from the report to a drop zone (e.g., onto a button for creating datasets)
- By dragging **pages** from the report to a drop zone (e.g., onto a button for creating datasets)
- By dragging **datasets** from the report to a drop zone (e.g., onto a button for creating datasets)
- By dragging **text** to a drop zone (e.g., onto a button for creating datasets)

Color highlighting is applied to indicate where the element can be dropped.

There are three possibilities, depending on how the project has been configured by the project developer:

- A new dataset is created as soon as the mouse button is released. If a medium, a page, a dataset, or a text has been selected, they will automatically be included in a suitable input element (assuming that the project developer has configured the project accordingly). Further data for the dataset (texts, images, etc.) can be entered subsequently (icon) and any placeholder information stipulated by the project developer can be adapted as necessary.
- The edit window containing the input elements that are available for the dataset concerned opens as soon as the mouse button is released. If a medium, a page, a dataset, or a text has been selected, they will – once again – already be preselected within a suitable input element (assuming that the project developer has configured the project accordingly). In the case of a dataset, the input elements may alternatively contain the content of the dropped dataset. The content of the edit window can be edited as required and then saved in the usual manner.

- The current project configuration does not allow the creation of a new dataset using drag-and-drop (message: "Unable to find any suitable templates").

For more information on creating datasets in ContentCreator, see also the FirstSpirit Manual for Editors (ContentCreator), chapter "Menu functions / Content area / Create dataset".

5.1.4.4 Links

When FirstSpirit version 5.1 was released it became possible to create links in the rich text editor (CMS_INPUT_DOM) and the rich text editor for tables (CMS_INPUT_DOMTABLE) not only by selecting the icon, but also – depending on the project configuration – by dragging and dropping elements into the editor. In FirstSpirit version 5.2, the link input types for creating **mouse-sensitive images** (CMS_INPUT_IMAGEMAP) are now also taken into account. Provided the project has been configured accordingly, this means that images (e.g., from search results in the Report area) can be dragged onto text that has already been entered in the editor. However, it is not possible to use images from the workstation in conjunction with this functionality. In the dialog for the newly created link ("Edit link"), the dropped image is used as a background image for the image map:

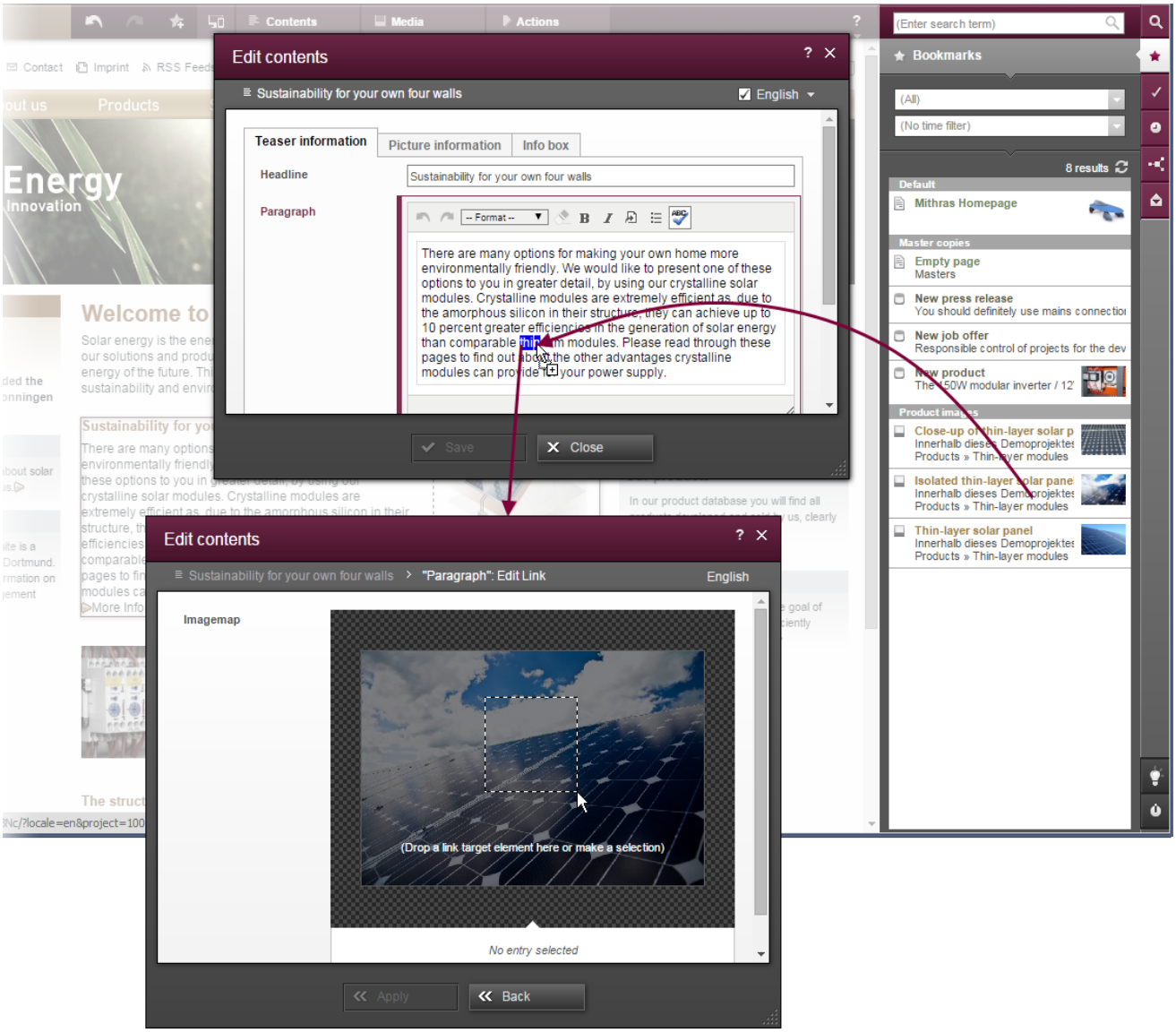


Figure -5-8: Creating a link with an image map

You can also use this dialog to create the frames and links for the image map.

In addition, links can now also be created in a rich text editor (CMS_INPUT_DOM) or in a rich text editor for tables (CMS_INPUT_DOMTABLE) by taking text from the address bar of a browser, from particular word processing programs, or from a web page (for example) and dragging it onto text in the rich text editor. Depending on the project configuration, the dropped text can be used either as a link text or as a link target. The rest of the information about the link can then be added by selecting "Edit link" from the context menu. If several link input types are available, the various options are displayed

when you drop the text into the editor:

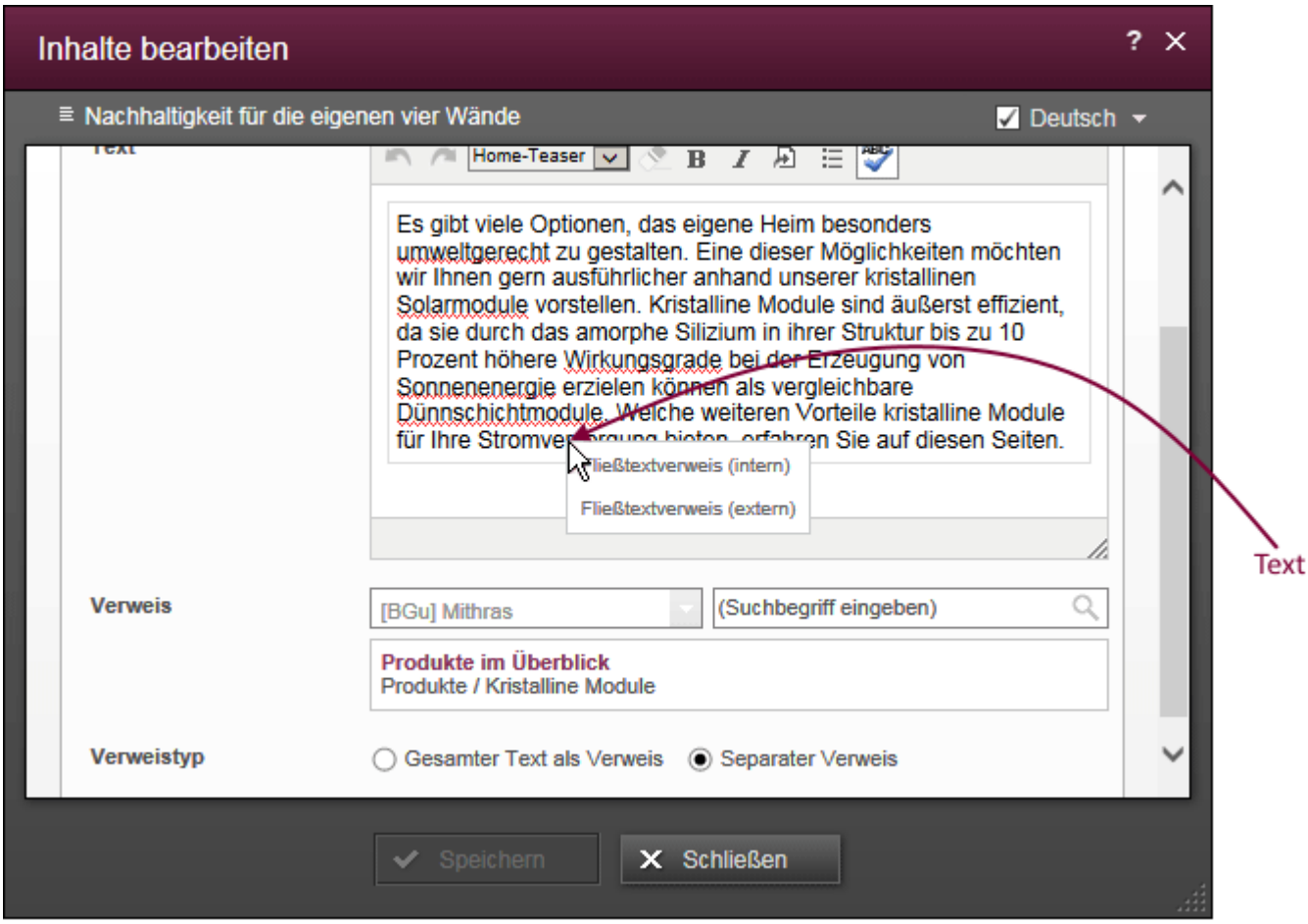


Figure 5-9: Several link input types available when dropping text

If you are using the "Google Chrome" browser and drop text from a word processing program into the rich text editor, the text may be removed from the external application[4]. This is not an error on the part of FirstSpirit!

For more information on working with links in general, on working with image maps, and on creating links using drag-and-drop, please also see the following chapters of the FirstSpirit Manual for Editors (ContentCreator):

[4] See https://productforums.google.com/forum/#!topic/chrome/HoQrTZzGh-U

- *"Editing a preview page / Input elements / Link input"*
- *"Editing a preview page / Input elements / Imagemap"*
- *"Editing a preview page / Input elements / Rich text editor", section "Insert/edit link"*
- *"Editing a preview page / Operating concept", section "Using drag-and-drop"*

5.1.5 Working with media in ContentCreator

5.1.5.1 Language-dependent media

It is now easier to manage media in ContentCreator. This includes the ability to store language-dependent media using ContentCreator. "Language-dependent" means that different media are displayed on multilingual websites depending on which language is selected. By contrast, it is also possible to have "language-independent" content (text, images, etc.) which – depending on how the project has been configured – is entered in ContentCreator input elements in one language only and is made available across all the different project languages. This may make sense, for example, when displaying images (without any text) or numbers (e.g., product descriptions, dimensions).

In light of the above, the following checkbox has now been added to the dialogs for uploading media (menu: "Media / Upload new medium") and for editing media (menu:"Media / Edit"): "Language-dependent: separate file for each language":

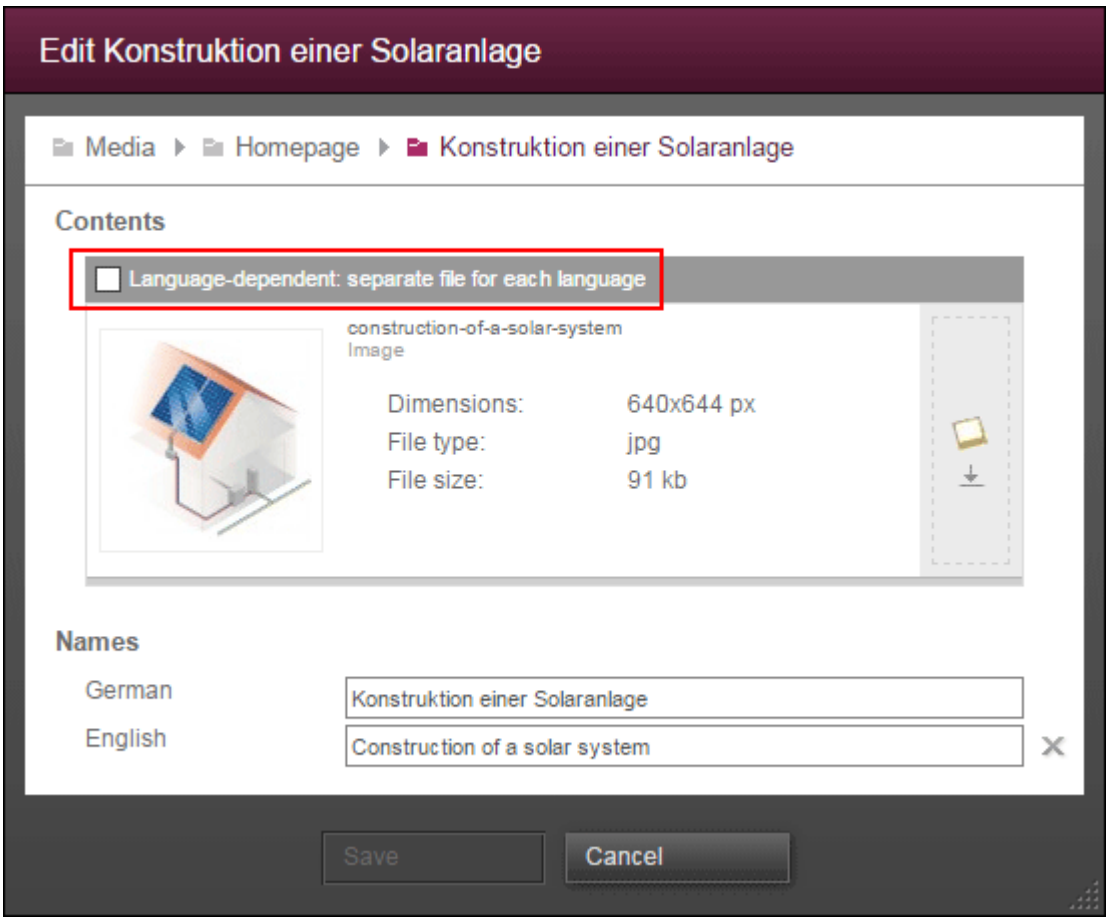


Figure 510: Editing a medium

- **In the "Upload new medium" dialog**, you can check this box if you want the new medium to be uploaded on a language-dependent basis. A selection list then appears on the right-hand side so that you can select which project language you want the medium to be uploaded to.
- **In the "Edit" dialog**, this checkbox indicates whether the medium concerned is language-dependent. On the right-hand side, there is a selection list for selecting which project language you want the changes to be applied to.

For more information on working with media, see the *FirstSpirit ContentCreator documentation*:

- "Menu functions / Media area / Upload media"
- "Menu functions / Media area / Edit medium"
- "Edit preview page / Media"
- "Menu functions / Media area / Manage media"

5.1.5.2 New "Manage media" dialog

The new "Manage media" dialog that can be accessed via the "Media" menu offers various functions, including:

- Upload media
- Edit media (rename, add language variant)
- Replace media
- Move media (to another storage location)
- Delete media (depending on project configuration)
- Add folder
- Rename folder
- Move folder
- Delete folder (depending on project configuration)

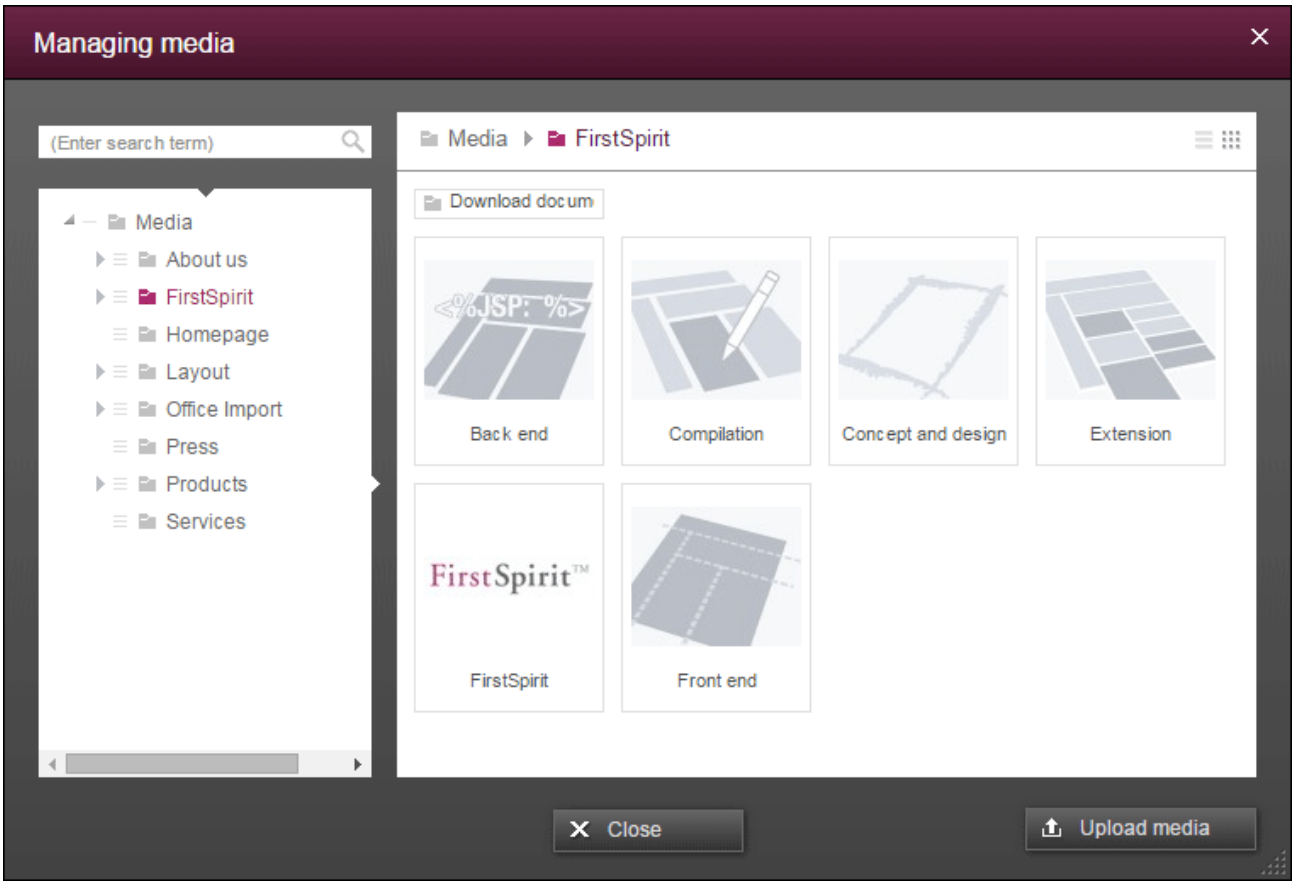


Figure 5-11: Media Store in ContentCreator

For more information on this dialog, see the *FirstSpirit ContentCreator documentation*, "Menu functions / Media area / Manage media".

5.1.6 Editing image variants

If the template developer has configured this feature, you can select image details or rotate or flip images displayed on the preview page as required. You can access this feature by selecting the  icon in the bottom right-hand corner of the referenced image.

In FirstSpirit, multiple variants of each image can be made available for use in different contexts, e.g., in different layouts or output media ("resolution"). Each image is generally available in a variety of sizes for use in different layouts and the system is capable of calculating the required size automatically.

Previously, it was only possible to edit one variant of an image in the "Edit image section" dialog (generally the one shown in the preview). However, as of FirstSpirit version 5.2, multiple variants can be edited, depending on what settings have been applied by the template developer.

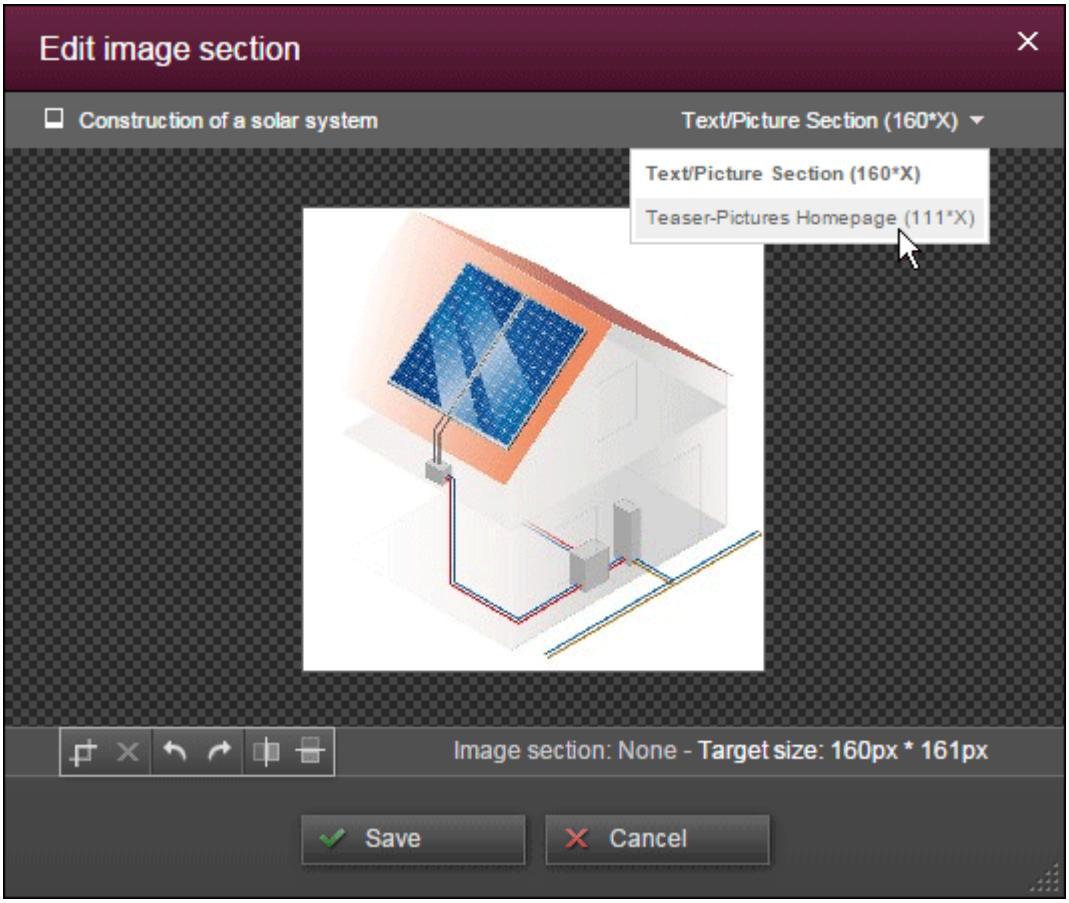


Figure 5-12: Image cropping – Multiple variants

In the top right-hand corner, you can see the name of the variant that can now be edited in the dialog. An arrow indicates that multiple variants can be edited. Click the arrow to display all the different variants that are available. The software indicates which image detail is currently selected for each variant. You can now select image details or rotate or flip images as required. Click "Save" to save the image in its current state and close the dialog.

Whenever an image variant is changed, all pages containing this image variant are updated with the edited image variant as well.

5.1.7 Editing forms from the comparison view

In ContentCreator, different states of a form can be placed side by side and compared.

To use this feature, click the  icon to access the difference display (function: "Display all changes" on pages with the state "Modified page", "In workflow", "Historic version" / "Comparison View").

The difference display compares the editorial changes on the current preview page with the last released state and highlights them accordingly. As of FirstSpirit version 5.2, corrections to the current state can now be made directly from this view. In addition, indicated changes can be reversed on a form-by-form basis.

For this purpose, the  and  icons are displayed below the "Release State" / "Current State" bar.

 **Edit**: This icon opens an edit window containing the input element concerned. In this window, you can change the content of the current state as required.

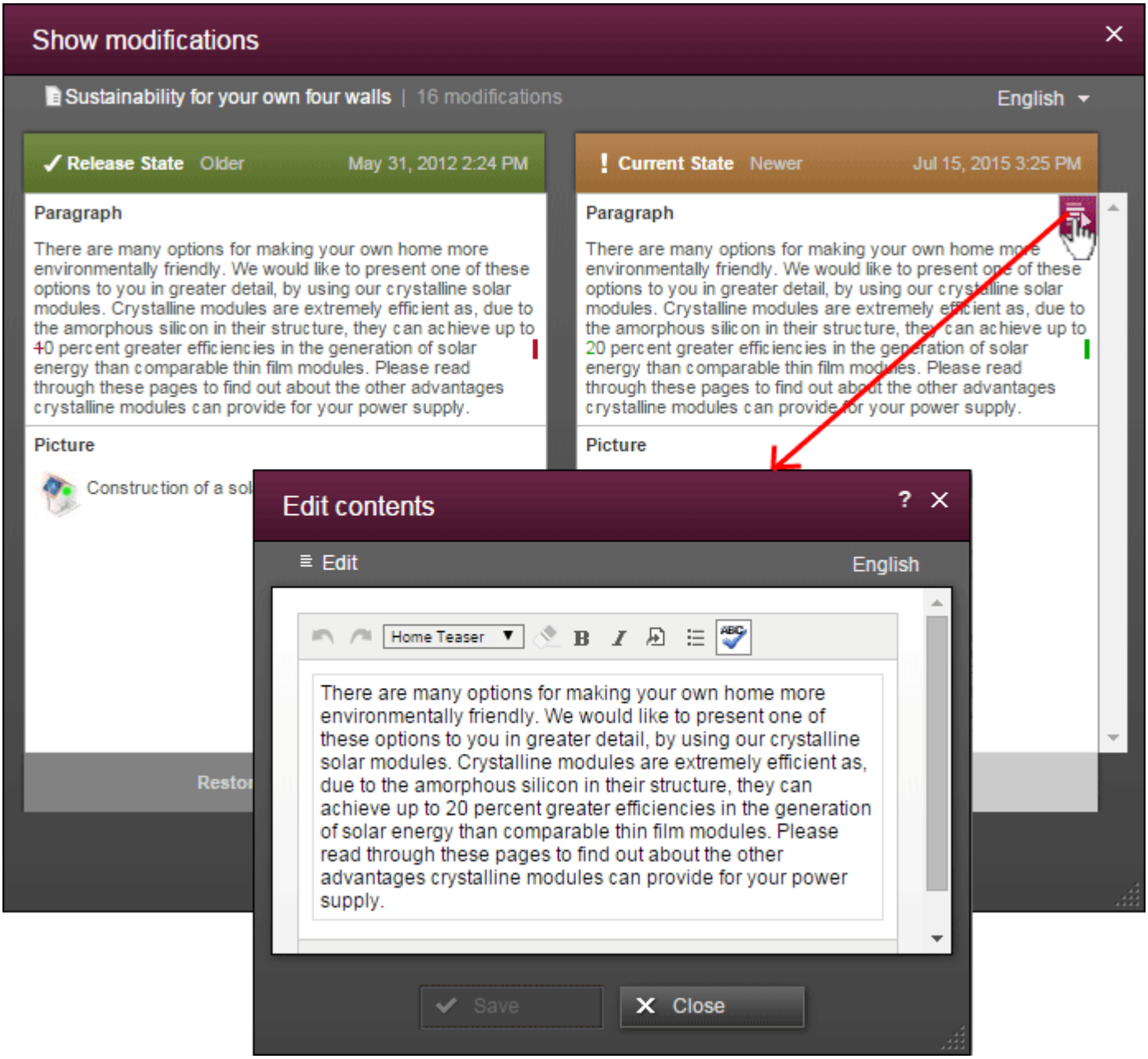


Figure 5-13: Edit icon in the difference display

Restore: This icon allows you to undo changes that were made previously and are visible as tracked changes in the left-hand column ("Release State"). When the subsequent confirmation prompt appears, click "OK" to revert to the version shown in the left-hand column.

For more information, see the *FirstSpirit ContentCreator documentation*, "Report area / Project history".

5.1.8 Optimizing work with nested list components (FS_LIST)

The list input component (FS_LIST) allows you to create and maintain lists. These can be lists of datasets, links, sections, or other types of data. Both the list itself and the individual entries are maintained in dedicated forms, with the list entries being maintained by means of "subforms" or "internal forms":

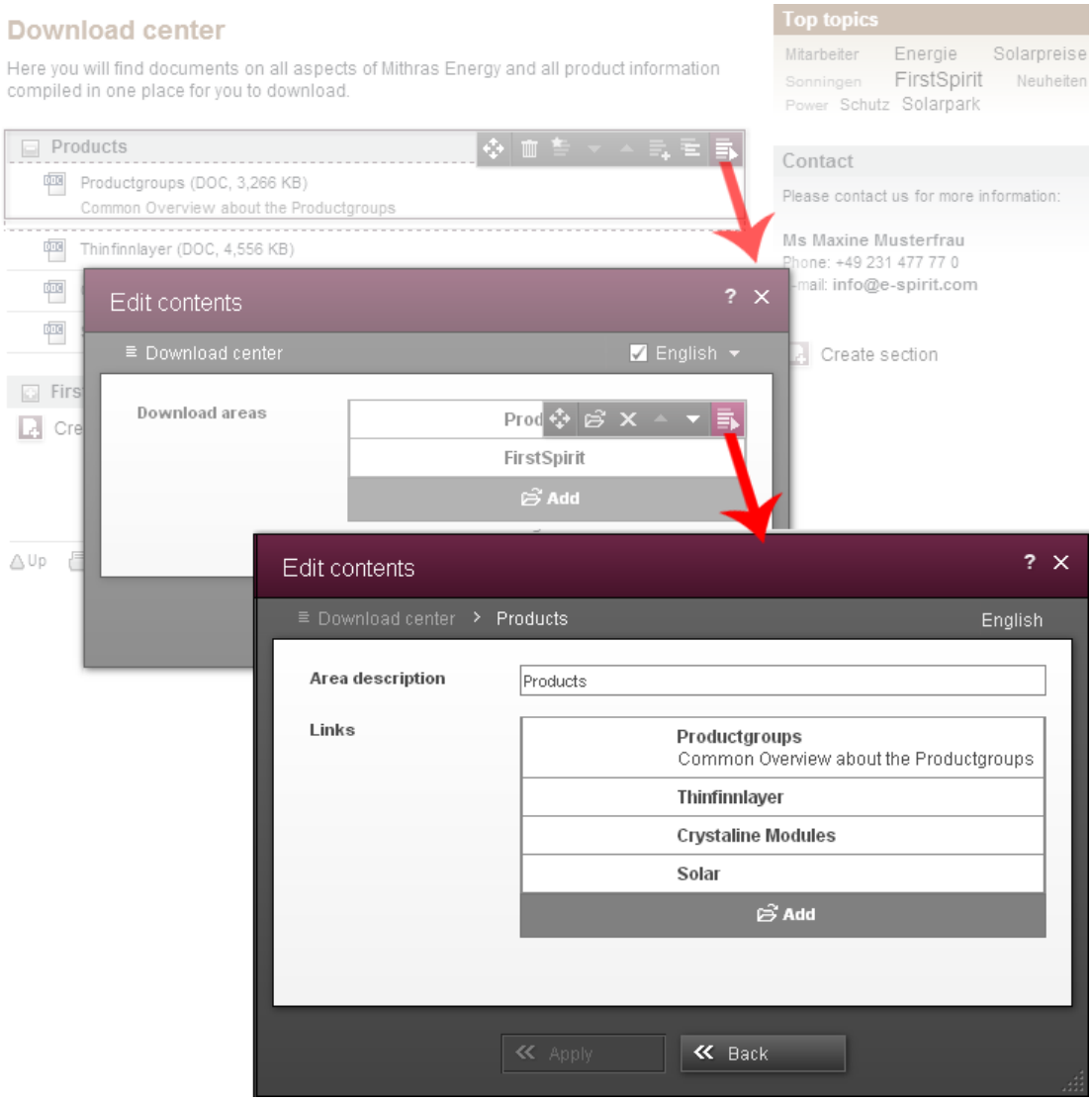


Figure 5-14: List input component (FS_LIST) with subform

This nesting is indicated by breadcrumb navigation in the title area of the respective form, e.g., in Figure 5-14:

Download center > Products

As of FirstSpirit version 5.2, you can switch to higher-level elements quickly and easily by clicking the breadcrumb entries. The breadcrumb can now be used for navigation even if other input elements for opening subforms are used (e.g., links in the rich text editor).

To help you find your way around more easily, the subdialogs also indicate which language can currently be used for maintaining the data. This is indicated in the top right-hand corner, but you cannot actually switch to a different language here. If you wish to enter data in another language, you must switch to a higher-level entry (e.g., using the breadcrumb navigation feature).

5.2 New/changed functions in SiteArchitect

5.2.1 Multi Perspective Preview in SiteArchitect

As Internet-enabled mobile devices such as notebooks, tablet PCs, and smartphones become more and more widespread, website designs need to be more and more flexible, with content which can be displayed perfectly on different display geometries and in different resolutions. That is why FirstSpirit version 5.1 introduced an easy way for editors to check what website content looks like and how well it can be navigated with a variety of display sizes in ContentCreator, while also allowing content, layouts, and images to be perfectly adapted for the output device concerned. Along with size considerations, other aspects can also be taken into account, e.g., previews for specific user groups ("Multi Perspective Preview", "MPP").

With the release of FirstSpirit version 5.2, the Multi Perspective Preview concept has been carried across to SiteArchitect as well.

Figure 5-15: Mobile content preview

Along with size considerations, other considerations can also be taken into account, such as the page's development over time (even in the future)

Figure 516: Time-dependent change to the content of a project

or previews for specific user groups. e.g.:

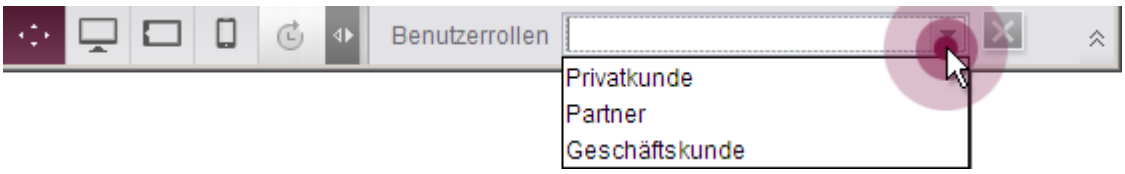


Figure 5-17: MPP – Example of user-specific perspective

To enable Multi Perspective Preview mode in SiteArchitect, go to the "MPP" tab in the integrated preview. It is operated in exactly the same way as in ContentCreator.

For detailed information, see the *FirstSpirit Online Documentation*, "Advanced topics / Multi Perspective Preview".

5.2.2 Unlimited preview tests

FirstSpirit version 5.2 introduces another way of testing content and layouts in various display sizes and for a variety of device and browser types. This test relies on a module and can be performed on a project-specific basis. Provided that the project has been configured accordingly, you can use the preview icon in the horizontal tool bar to display various submenu items which allow you to see what the content looks like in the AppCenter area with a wide range of browsers and device types, as well as in various versions and with different operating systems (e.g., Windows, Mac OS, Android, etc.). To access this function, simply click the downward pointing arrow and select the "Plug-ins" entry.

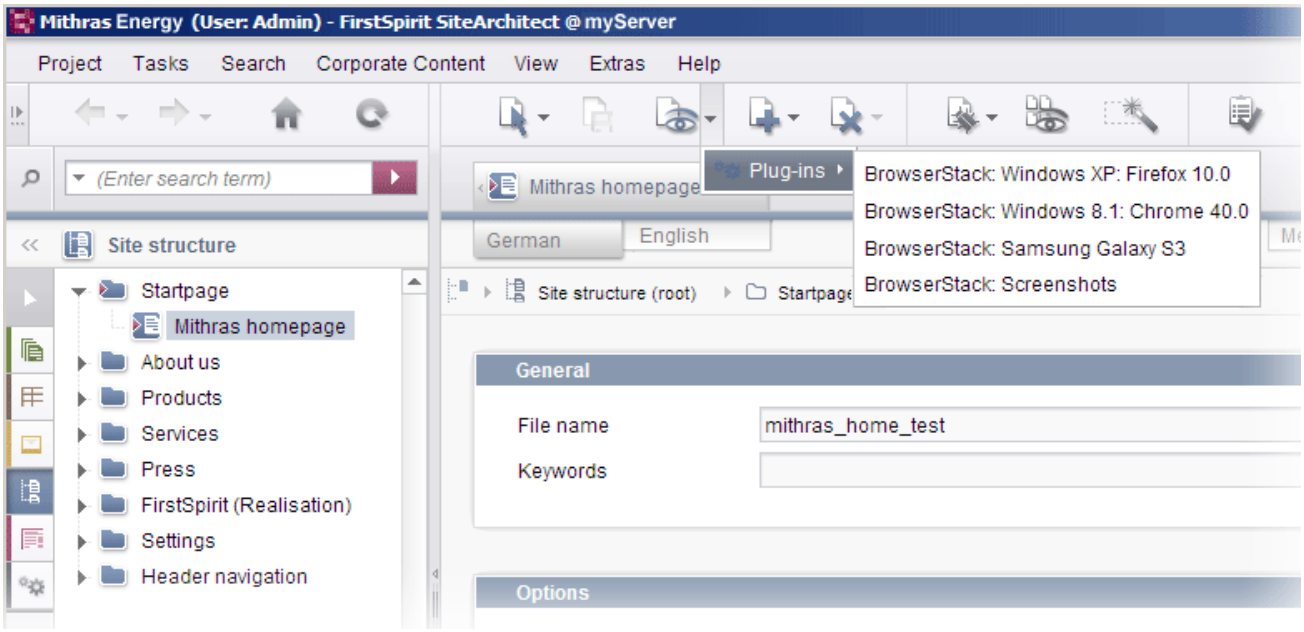


Figure 5-18: Cross-browser and cross-platform testing

Depending on how the project has been configured, it may be possible to integrate other plug-ins via the preview icon as well.

5.2.3 Working with metadata

As of FirstSpirit version 5.2, you can tell whether or not metadata has been specifically set for the current node simply by referring to the "Metadata" tab in the workspace.

"Metadata set" mode:

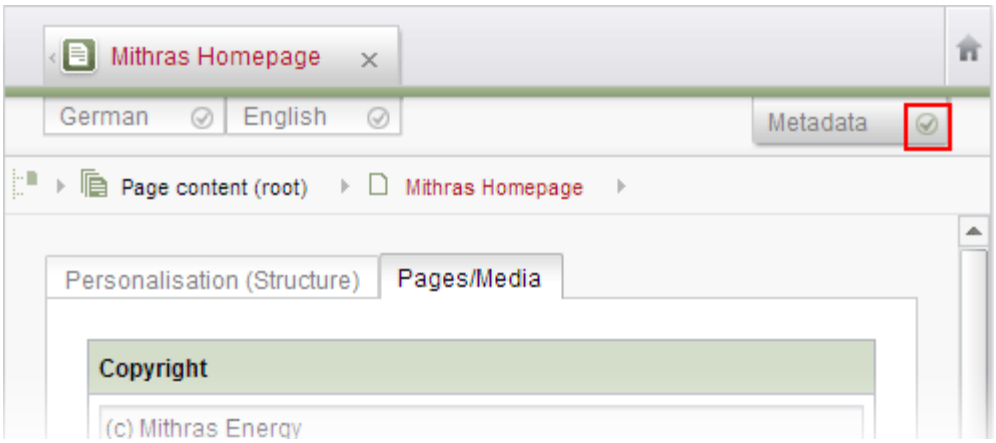


Figure 5-19: Metadata tab on a page in the Content Store

If metadata has been specifically set, a check mark appears on the "Metadata" tab.

If the node concerned is in Edit mode, all the metadata for the node can be deleted by removing the check mark with a click of the mouse (after confirming the prompt).

"Metadata not set" mode:

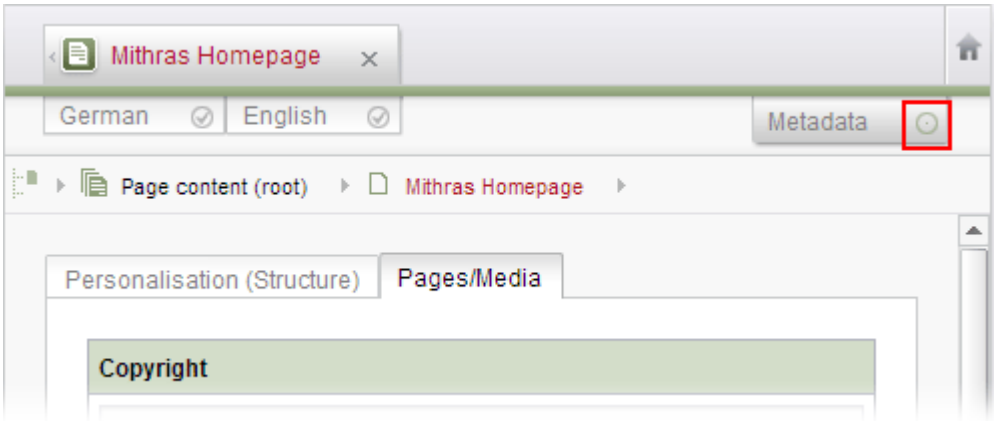


Figure 5-20: No metadata set

If no specific metadata is set, no check mark appears.

In this mode, you cannot edit the fields of the form on the metadata tab even if Edit mode has been enabled for the node concerned.

However, inherited metadata is still displayed in this mode, i.e., metadata that has been specifically set for higher-level nodes. To determine which higher-level node is acting as the source of the metadata displayed for the current node, you need only look for the i icon in the tree structure (go to the "View" menu and enable the "Show symbols (Metadata, Packages, Permissions) entry).

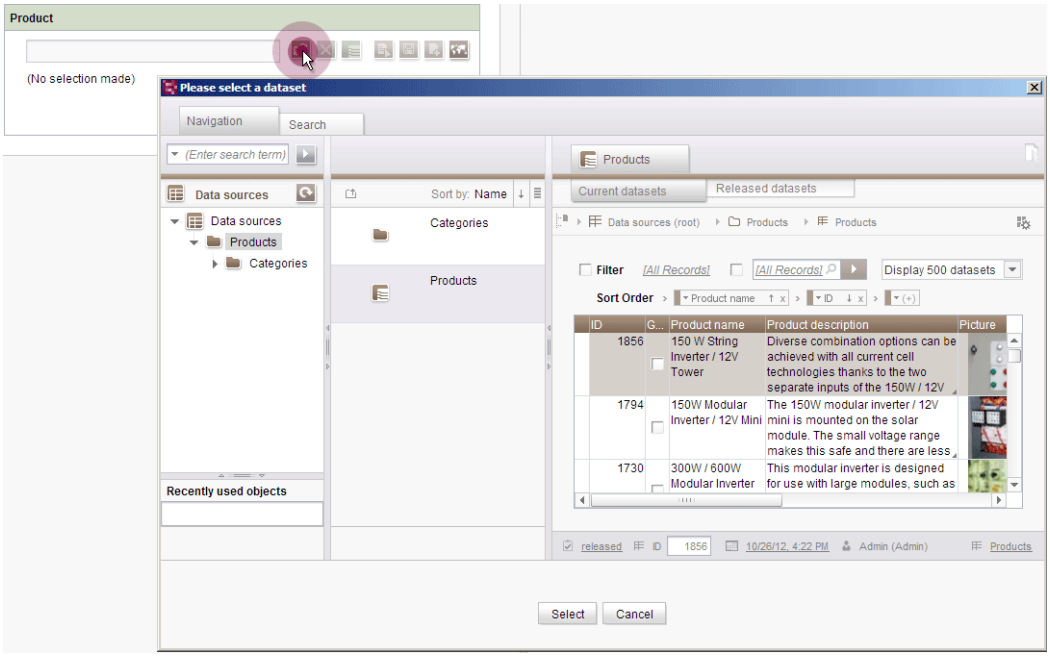
If you want to set metadata, you must switch to the "Metadata set" mode (see above) by clicking with the mouse so that the check mark appears. You can only do this if the node is in Edit mode.

For more information on working with metadata in SiteArchitect, see also: FirstSpirit SiteArchitect Manual, chapter "Metadata"; FirstSpirit Online Documentation, area: "Template development" / "Variables" / "Definition and output" / "in metadata".

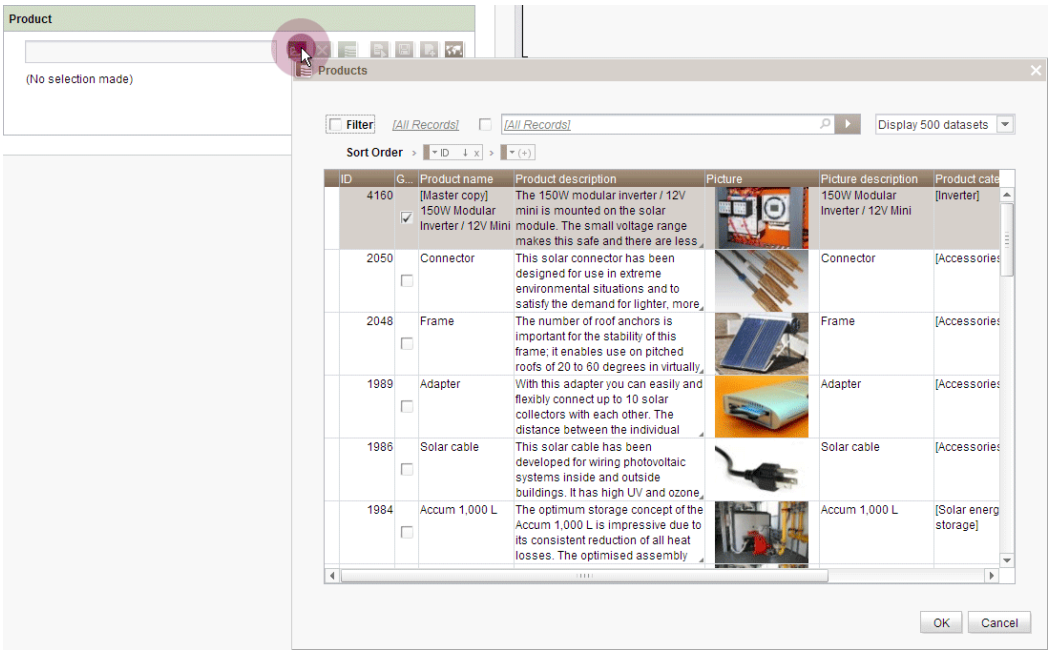
5.2.4 FS_DATASET: New selection mode

Where datasets in FS_DATASET in SiteArchitect were previously selected via a selection dialog, they can now be selected from a simple list.

Selection via dialog:



Selection from list:



5.2.5 Continuously adjustable enlargement and reduction of input components

As of FirstSpirit version 5.2, the height of the CMS_INPUT_DOM (rich text editor), CMS_INPUT_DOMTABLE (table), and FS_LIST (list) input components is continuously adjustable. For this purpose, an adjustment handle has been added in the bottom right-hand corner.

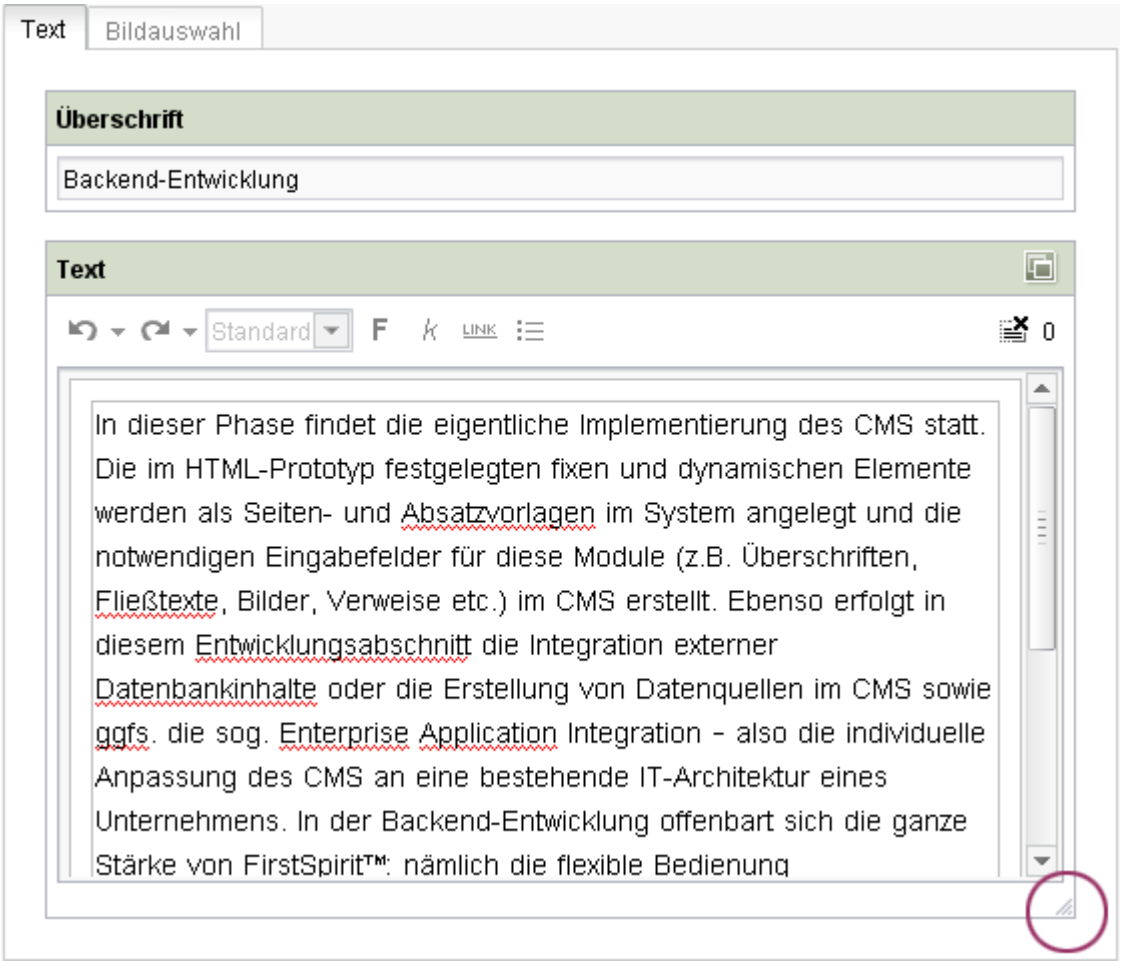


Figure 5-21: CMS_INPUT_DOM

The height of the input components cannot be reduced below the height value specified via the `rows` (CMS_INPUT_DOM, CMS_INPUT_DOMTABLE) or `height` (FS_LIST) parameters.

The previous icons for enlarging/reducing the size  no longer apply in this context.

5.2.6 New features in the Media Store

In FirstSpirit, images are stored in the Media Store. The project developer can specify various image sizes which then apply for all images in the project. In the language of FirstSpirit, these image sizes are called "resolutions". An image 450 x 301 pixels in size that is uploaded in FirstSpirit can be used for the main text column, for example, but an image to be used as a teaser image in a margin column must only be 117 pixels wide. FirstSpirit automatically scales the images to the defined resolutions (to 117 pixels wide,

for example). As a result, the uploaded image is then available in the Media Store both in the original size in which it was uploaded ("original resolution") and in the defined resolutions. This provides an easy way of standardizing the sizes of images for presentation that are provided in different original sizes. As an alternative to resizing an image, its content can be cropped or an entirely different image can be uploaded, according to prevailing requirements.

The functions for cropping or for selecting an alternative image for a resolution can be found under "Resolutions" in the workspace of an image in the Media Store in SiteArchitect.

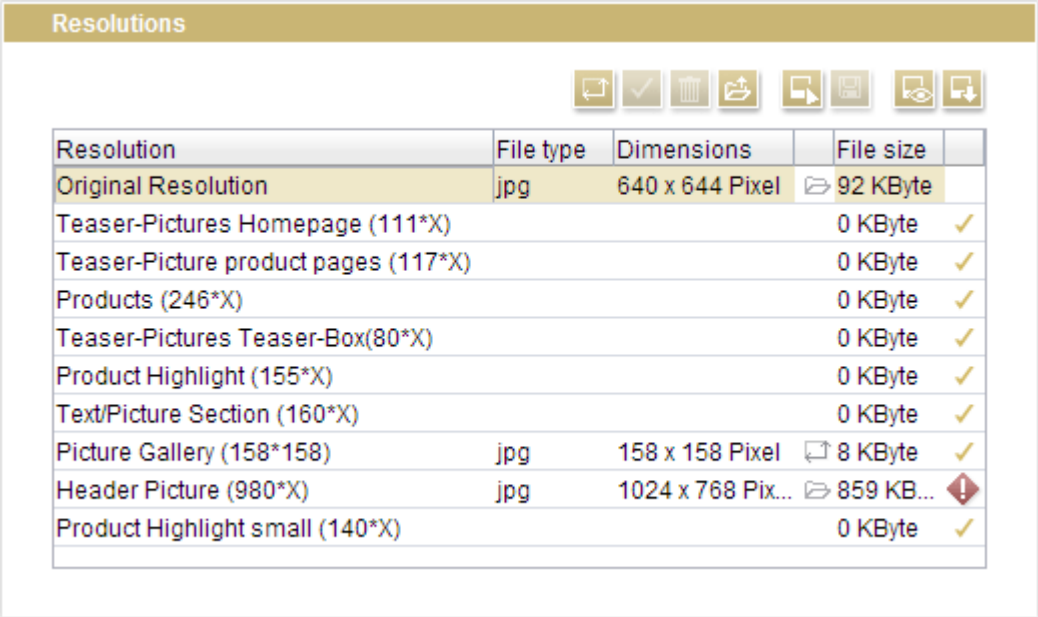


Figure 5-22: Image - Resolutions - List view edit area

Download: Click this new icon to download a copy of the selected resolution in the relevant file format to the local file system on the workstation. The required download folder can be selected via the file selection dialog.

A number of icons and buttons in this area have also been redesigned.

5.2.7 Error reporting enhancements

Software development is a cyclical process involving multiple phases. Aside from the development and implementation of new functions, the interrelated phases of quality assurance and debugging are another essential part of this. In version 5.2, a conscious

effort has been made to make error reporting easier and more comprehensive so that information about potential software bugs can be obtained from FirstSpirit while it is actually in use and then sent to the manufacturer in a form that makes it quick to analyze and process. This will play a major role in enabling us to fix bugs more quickly, thereby helping to improve the quality of the software.

FirstSpirit offers a special infrastructure for collecting errors and exceptions. A loader icon is displayed in the bottom left area of SiteArchitect for this purpose; it continually shows data transmission during editing work. 

In the event of an error message, a number is added to the loader icon to indicate how many error messages have occurred. Additional information on the error that occurred can then be requested by clicking on the icon. An information dialog appears showing an overview of current error messages as well as older ones.

In FirstSpirit version 5.2, this information dialog has been extended:

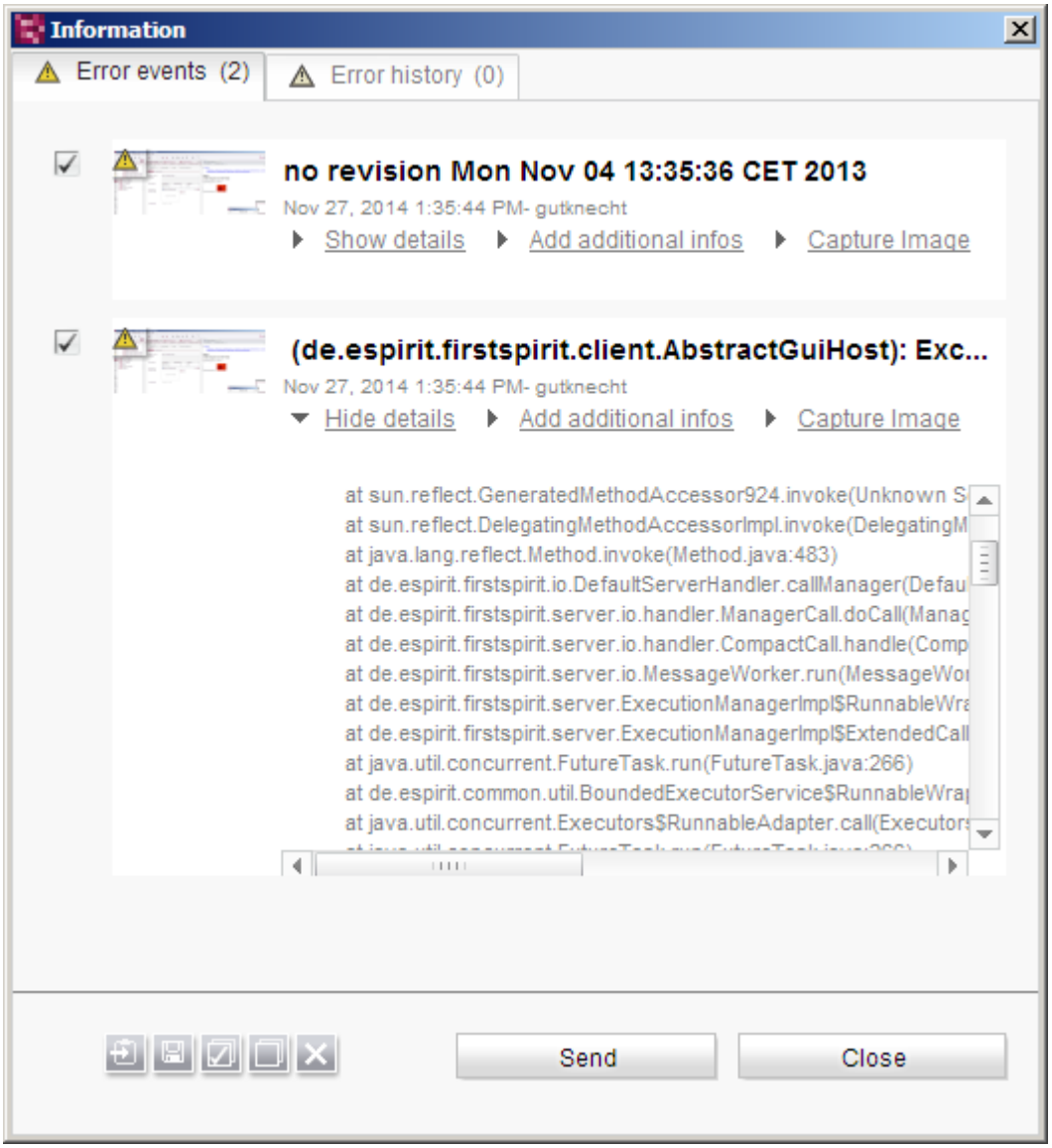


Figure 5-23: Centralized error collection in FirstSpirit SiteArchitect

Add additional information: Click this link to open an area for adding extra information about the error, e.g.:

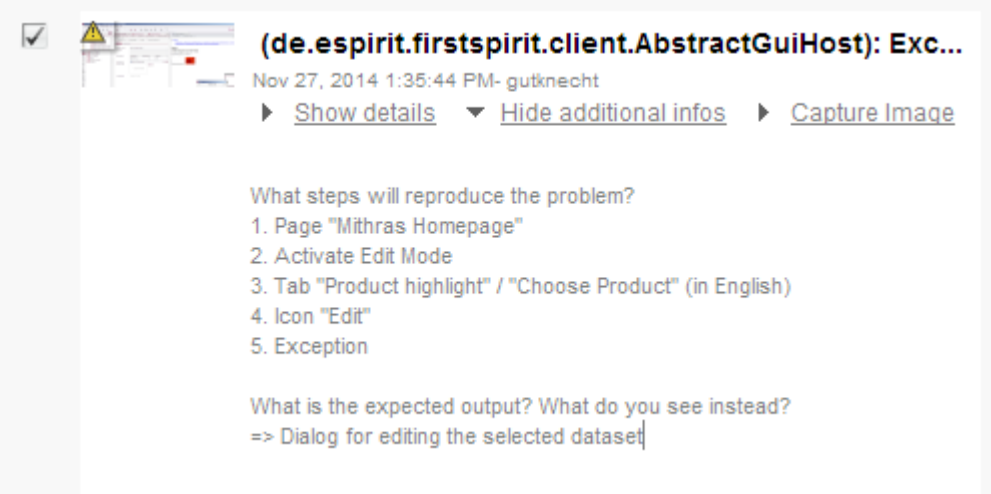


Figure 5-24: Adding additional information

The default text can either be utilized or removed as required.

Capture image: Clicking this link switches the focus to SiteArchitect. You can take a screenshot by pressing <ALT> and clicking with the mouse. To switch back to the dialog (Figure 5-23) without taking a screenshot, press <ESC>.

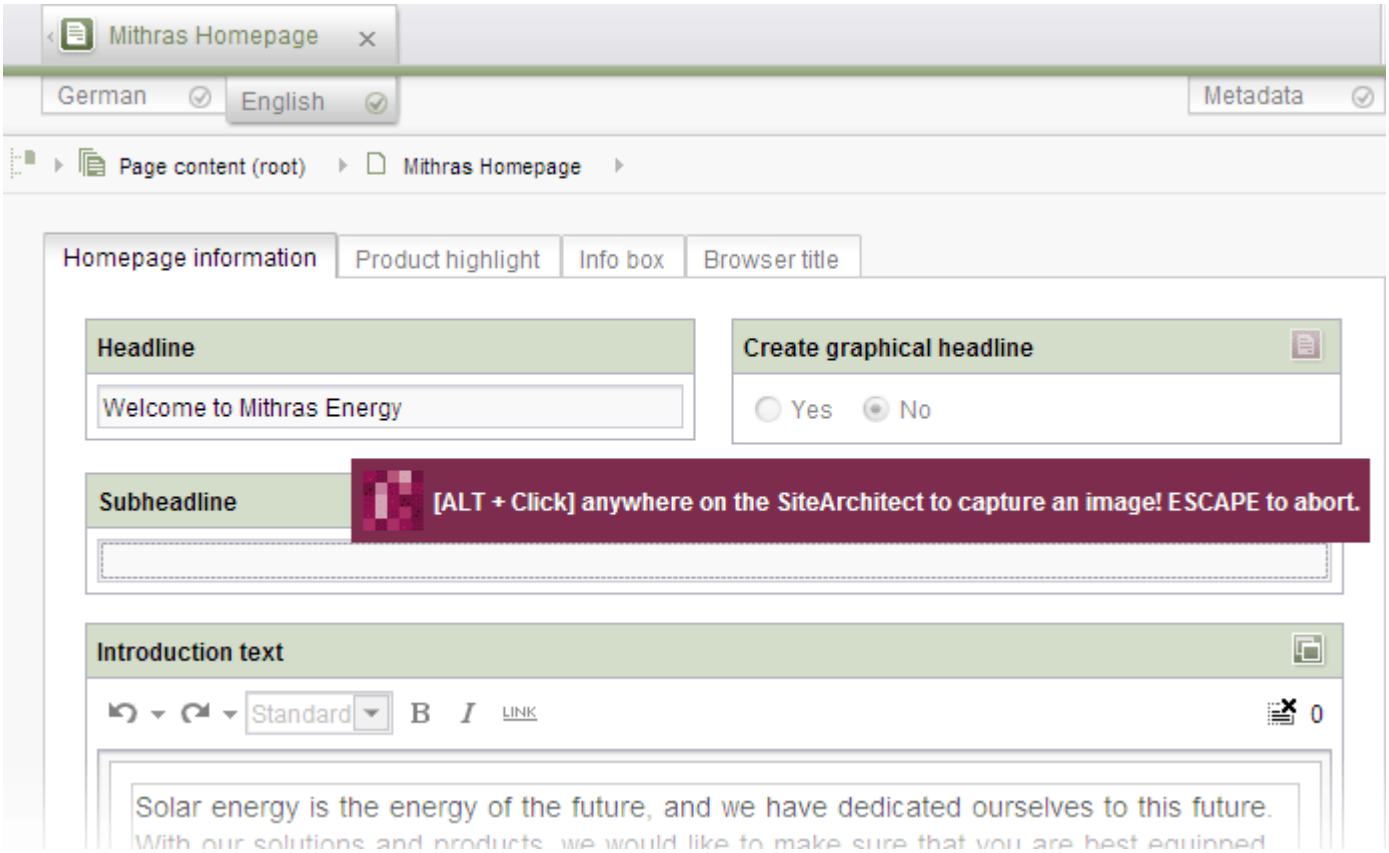


Figure 5-25: Taking a screenshot

Send: Click this button to send the selected error messages to the FirstSpirit server along with the supplementary information that has been added manually and your screenshot in an error report. It can also be transferred automatically to the manufacturer e-Spirit. This function must have been configured accordingly by the FirstSpirit administrator.

For more information on the error collector in SiteArchitect, see also the *FirstSpirit SiteArchitect documentation*, chapter "Displaying errors".

6 New functions for template developers

6.1 New input components: FS_CATALOG and FS_INDEX

As part of FirstSpirit version 5.2, two new input components have been introduced which support functions for bundling content. FS_CATALOG and FS_INDEX have been introduced to simplify configuration and improve usability, particularly as far as nested components are concerned. The operating concept has been inspired by the design and functionality of the data store, which means that editors always edit internal elements in the (central) workspace and can see an overview of the existing entries on the left-hand side of the screen:

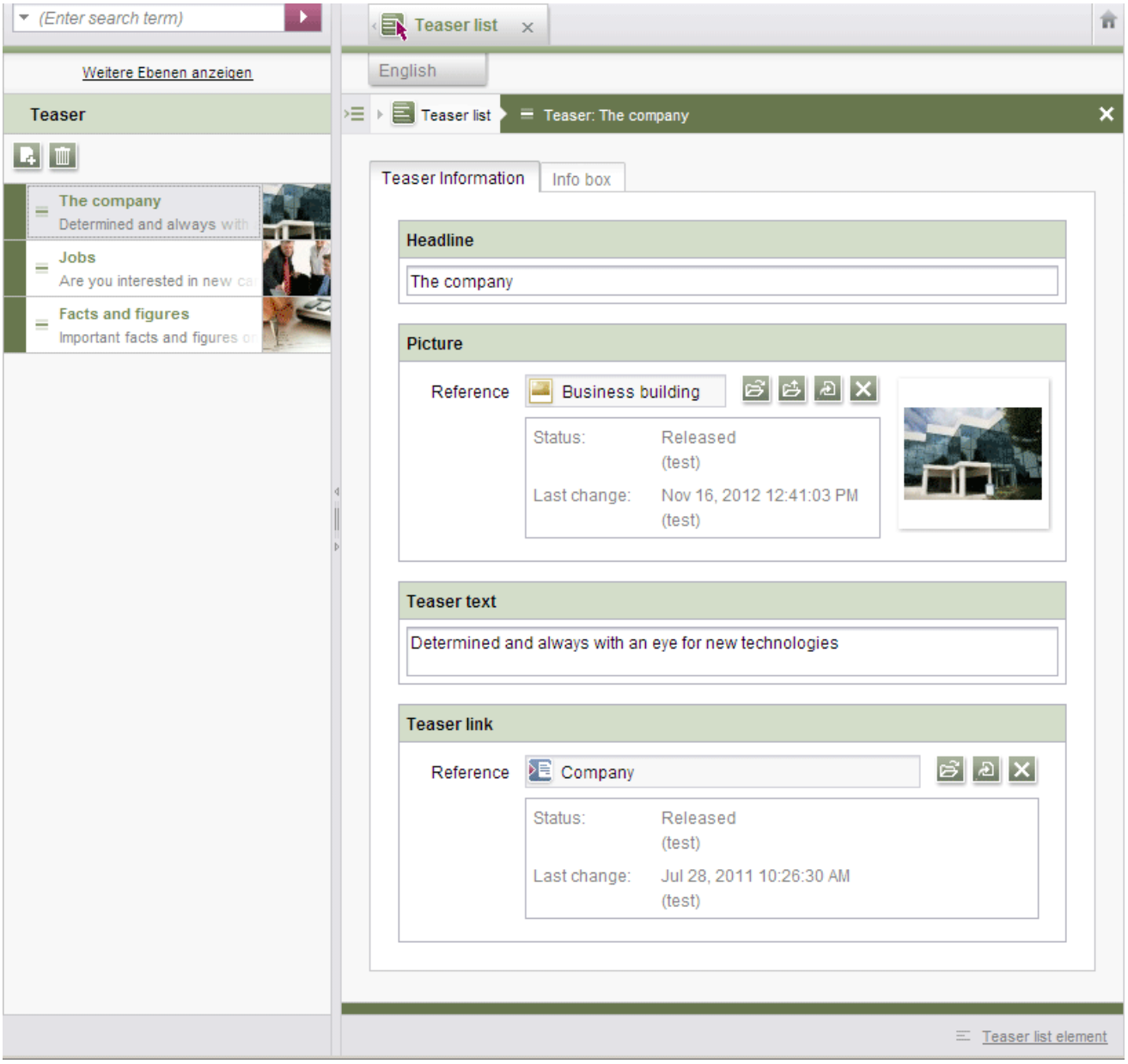


Figure 6-1: FS_CATALOG with navigation

For more information, see also chapter 4.3, page 15.

6.1.1 Syntax

Example for **FS_CATALOG**:

```
<FS_CATALOG name="fs_catalog_sections" useLanguages="no">
    <LANGINFOS>
      <LANGINFO lang="*" label="Section list"/>
      <LANGINFO lang="DE" label="Absatzliste"/>
    </LANGINFOS>
    <TEMPLATES type="section"/>
</FS_CATALOG>
```

Example for **FS_INDEX**

```
<FS_INDEX name="st_index">
        <LANGINFOS>
          <LANGINFO lang="*" label="Dataset selection"/>
          <LANGINFO lang="DE" label="Datensatzauswahl"/>
        </LANGINFOS>
        <SOURCE name="DatasetDataAccessPlugin">
          <TEMPLATE uid="Products.products"/>
        </SOURCE>
</FS_INDEX>
```

Explanation:
The data source is specified using the *SOURCE* tag. The *name* attribute is used to specify the reference name of the DataAccess plug-in.

The use of a DataAccess plug-in that implements the "Reporting" aspect in the client requires an AppCenter license. See also the FirstSpirit Online Documentation, "Plug-In Development / Universal Extensions / Data Access / Use for Reporting" and the FirstSpirit Manual for Administrators.

As standard, FirstSpirit includes a solution for selecting datasets from project data sources (see Figure 4-4). The name of the associated DataAccess plug-in is *DatasetDataAccessPlugin*. The required table template must also be specified with `TEMPLATE / uid`.

Datasets with a GID ("FS_GID" column) are a mandatory requirement when using the DatasetDataAccessPlugin. See also the FirstSpirit Documentation for Administrators, "Database connection" chapter and FirstSpirit release notes for version 5.1, "Switching from older FirstSpirit versions" chapter.

See also chapter 6.13.3.8, page 121, "de.espirit.firstspirit.client.plugin.dataaccess".

For documentation on tags and parameters for defining forms and output methods, see the following pages of the *FirstSpirit Online Documentation*:

- "Template development / Forms / Input components / CATALOG"
- "Template development / Forms / Input components / INDEX"
- "Template development / Template syntax / Data types / Catalog"
- "Template development / Template syntax / Data types / Card"
- "Template development / Template syntax / Data types / Index"
- "Template development / Template syntax / Data types / Record"

6.1.2 Migration

The **FS_CATALOG** input component can receive and process values that have been entered using input components of the type FS_LIST, INLINE. If the identifier of the input component specified with the `name` parameter is retained and the syntax of FS_LIST is converted to FS_CATALOG, data that has previously been entered by the editor will be transferred, can continue to be saved, and will be editable.

Example:

```
<FS_LIST name="st_sections"...>
```

must be replaced with

```
<FS_CATALOG name="st_sections"...>
```

For details of which tags and parameters are supported by FS_CATALOG, see the *FirstSpirit Online Documentation*.

The New, Delete and Edit functions can be deactivated by means of rules:

- NEW: New entry
- REMOVE: Remove entry
- EDIT: Edit entry

Example:

```
<RULES>
  <RULE when="ONLOCK">
    <WITH>
      <FALSE/>
    </WITH>
    <DO>
      <PROPERTY name="NEW" source="fs_catalog"/>
    </DO>
  </RULE>
</RULES>
```

*The software does **not** support a "downgrade" to FS_LIST: Once data has been saved with FS_CATALOG, it can no longer be converted back to FS_LIST.*

FS_INDEX is not compatible with the FS_LIST input component in terms of the storage format used. However, this does not apply if data is linked using a foreign key relationship, as this is just as easy for the system data object source to understand. Aggregated relationships are subject to a restriction in that their datasets are treated as independent content objects as well (no automatic process of simultaneous validation on saving or on release).

FS_CATALOG / FS_INDEX deliberately do not function in the same way as FS_LIST.

Currently, there is no replacement available for FS_LIST, Page type.

6.2 Template wizard

Most of the concept, design, and layout for a new website is developed externally by an agency. The HTML templates (mockups) supplied are then transferred to a FirstSpirit project via the FirstSpirit Template Store.

To optimize the developer experience, the FirstSpirit infrastructure partially automates the process of transferring HTML mockups to FirstSpirit. It provides functions which can analyze the HTML templates supplied, identify referenced images and files in HTML content, and import all required content into FirstSpirit. To do this, FirstSpirit SiteArchitect provides a graphical user interface in the shape of the template wizard.

The template wizard guides the template developer through the process of generating FirstSpirit templates in just a few steps:

- Import of supplied HTML and JSP structures (locally or via URL), automatic transfer of images and files from the HTML mockup ("auto-import")
- Automatic transfer of design and layout specifications ("auto-import")
- Form builder, which creates the corresponding FirstSpirit input components for the HTML content supplied, e.g., a DOM editor for an editorial text or an FS_REFERENCE type component for images referenced in the text
- Assignment of form elements to FirstSpirit templates
- Assignment of content areas to page templates

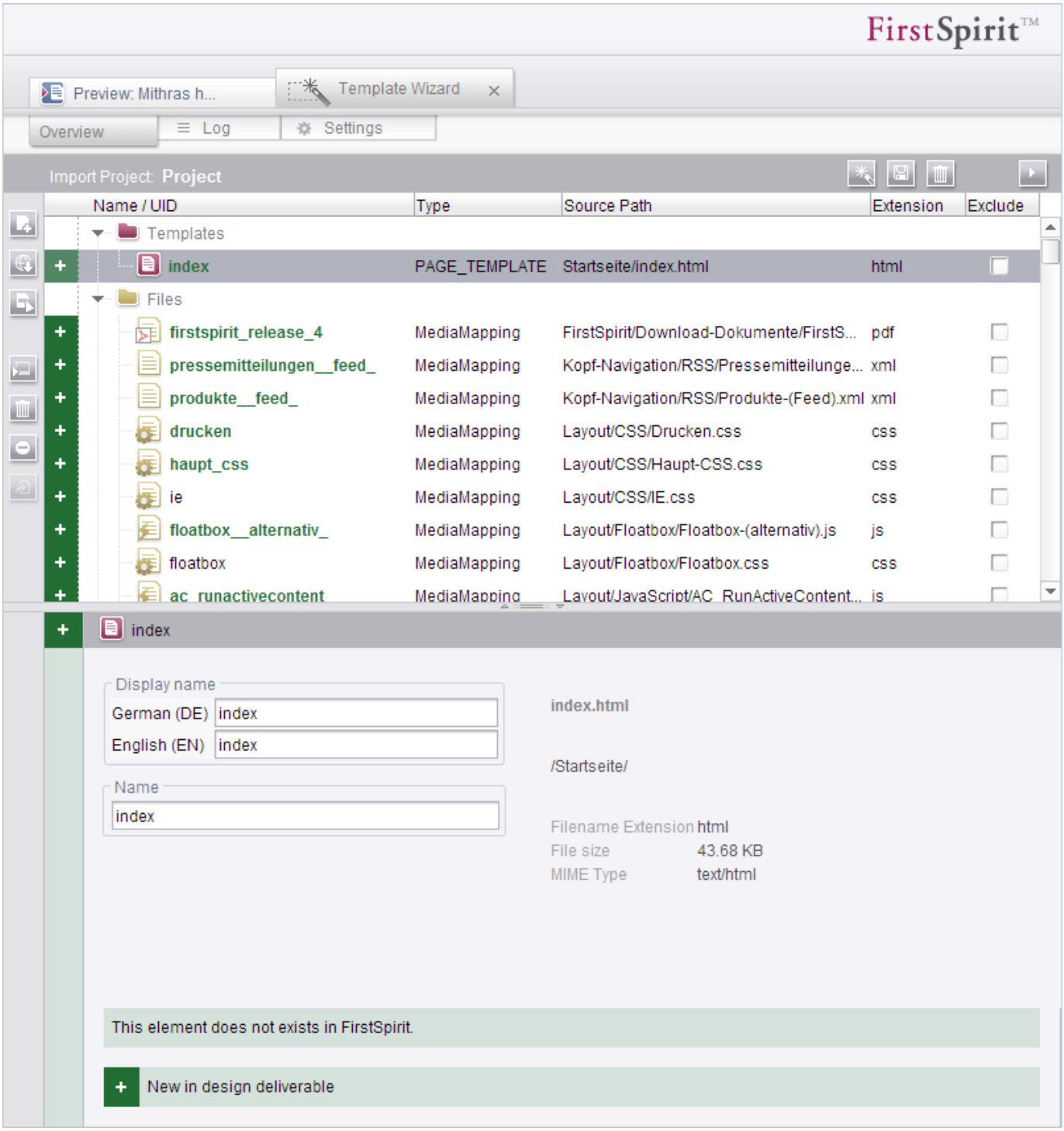


Figure 6-2: Template wizard

Following completion of the graphics-based import and analysis phase in the template wizard, the results can be imported into the corresponding subareas (templates, media)

of the FirstSpirit project.

The time and effort required from delivery of HTML mockups to creation of the initial project prototype can be significantly reduced.

The template wizard can be started by clicking the "Template Wizard" icon on the tool bar in FirstSpirit SiteArchitect.

For detailed information, see the *FirstSpirit Online Documentation*, "Template development / Template wizard".

6.3 Enhancements for the "Rules" area ("Dynamic forms")

The syntax for the "Rules" area ("Dynamic forms") has been optimized and extended. For more information, see also the *FirstSpirit Online Documentation, "Template development / Rules"*.

Existing rule definitions from earlier FirstSpirit versions are still valid in FirstSpirit version 5.2 and will be interpreted accordingly. However, the old syntax will become invalid and will cease to be supported at some point in the future. Therefore, the new syntax should be adopted as soon as possible. See also chapter 9.3, page 146.

6.3.1 <RULE/> tag

Each rule is enclosed by an opening and a closing `<RULE/>` tag.

```
<RULES>
     <RULE>
           [Rule 1]
     </RULE>
     <RULE>
           [Rule 2]
     </RULE>
</RULES>
```

You also have the option of including the `when` attribute for each rule (see chapter below).

6.3.2 Execution time (`when`)

The optional `when` attribute can be used to specify that the respective rule:

- Should be executed once on saving: `when="ONSAVE"`
- Should be executed once when the user switches to edit mode or creates a new object (page, section, dataset):`when="ONLOCK"`

Example:

```
<RULES>
     <RULE when="ONLOCK">
  ...
```

By default (i.e., if the `when` attribute has not been specified), the rule is executed whenever the editor makes an entry.

6.3.3 Restriction levels (`INFO`, `SAVE`, `RELEASE`)

The restriction levels that were previously defined by means of the

- `<ON_EVENT/>`
- `<ON_SAVE/>`
- `<ON_RELEASE/>`

tags are now mapped via the `scope` parameter in the `VALIDATION` tag. These parameters can be set to the following values:

- `INFO`
- `SAVE`
- `RELEASE`

The default setting (i.e., if the `scope` parameter has not been specified) is `INFO`.

Example of old approach (up to and including FirstSpirit version 5.1):

```
<ON_SAVE>
  <VALIDATION>
        <PROPERTY name="VALID" source="st_startDate"/>
        <MESSAGE lang="*" text="Start date must be set"/>
        <MESSAGE lang="DE" text="Startdatum muss gesetzt werden"/>
  </VALIDATION>
</ON_SAVE>
```

Example of new approach (as of FirstSpirit version 5.2):

```
<VALIDATION scope="SAVE">
        <PROPERTY name="VALID" source="st_startDate"/>
        <MESSAGE lang="*" text="Start date must be set"/>
        <MESSAGE lang="DE" text="Startdatum muss gesetzt werden"/>
</VALIDATION>
```

6.3.4 Setting input components to an empty value (EMPTY)

Input components can contain a value or be left "empty". If no value is stored in the input component, this can return either an

- empty value (e.g., CMS_INPUT_TEXT) or a
- NULL (e.g., CMS_INPUT_NUMBER)

depending on the type of input component concerned.

The EMPTY property allows you to

- check whether a value has been stored (or pre-assigned in the template) for an input component or whether it is empty, e.g.:

```
<WITH>
        <PROPERTY name="EMPTY" source="gadget"/>
</WITH>
```

- and also to set an input component to an empty value, e.g.:

```
<DO>
        <PROPERTY name="EMPTY" source="gadget"/>
</DO>
```

As of FirstSpirit version 5.2, an EMPTY check can now be performed for all the different input component types.

For more information, see the FirstSpirit Online Documentation, "Template development / Rules / Form properties <PROPERTY/> / Property EMPTY ".

6.3.5 Null check (NULL)

In previous versions, you could only check whether an input component contained a value by carrying out a "not null check" (using the `<NOT_NULL/>` tag). With the release of FirstSpirit version 5.2, you can now perform null checks as well. This means that you can use the `<NULL/>` tag (as well as `<NOT_NULL/>`) for value determination within the rule definition (`<VALIDATION/>` tag) or when defining a precondition (`<IF/>` tag). If the input component contains a value, the `<NULL/>` check returns FALSE and if it does not contain a value, the `<NULL/>` check returns TRUE.

For more information, see the FirstSpirit Online Documentation, "Template development / Rules / Comparative expressions / <NULL/>/<NOT_NULL/>".

6.3.6 Negation of Boolean values (NOT)

Previously, the only way to negate Boolean values from value determination (`<WITH/>` or `<SCHEDULE/>`) was to perform the negation within the value determination itself. With the release of FirstSpirit version 5.2, it can now also be performed in the `<DO/>` tag. This involves using the `<NOT/>` tag in conjunction with the following tags:

- `<PROPERTY/>`
- `<VALIDATION/>`
- `<NOT/>`

Examples:

With `<PROPERTY/>`:

```
<RULES>
   <RULE>
      <WITH>
          <PROPERTY name="EMPTY" source="st_text_1"/>
      </WITH>
      <DO>
          <PROPERTY name="VISIBLE" source="st_text_2"/>
          <NOT>
              <PROPERTY name="VISIBLE" source="st_text_3"/>
          </NOT>
      </DO>
   </RULE>
<RULES>
```

In this example, the template contains three text fields (text field 1: "st_text_1", text field 2: "st_text_2", text field 3: "st_text_3"). Text field 1 is always displayed, text field 2 is displayed if text field 1 is empty, and text field 3 is displayed if text field 1 is not empty.

With `<VALIDATION/>`:

```
<RULES>
   <RULE>
      <WITH>
          <PROPERTY name="EMPTY" source="st_headline"/>
      </WITH>
          <DO>
              <NOT>
                 <VALIDATION scope="SAVE">
                 <PROPERTY name="VALID" source="st_headline"/>
                  <MESSAGE lang="*" text="The editor must not be
                                          empty!"/>
                  <MESSAGE lang="DE" text="Der Editor darf nicht
                                          leer sein!"/>
                 </VALIDATION>
              </NOT>
          </DO>
   </RULE>
</RULES>
```

With <NOT/>:

```
<RULES>
   <RULE>
      <WITH>
          <PROPERTY name="EMPTY" source="st_text"/>
      </WITH>
      <DO>
          <NOT>
              <NOT>
                  <VALIDATION>
                       <PROPERTY name="VALID" source="st_text"/>
                       <MESSAGE lang="*" text="Nicht leer!"/>
                  </VALIDATION>
              </NOT>
          </NOT>
      </DO>
   </RULE>
</RULES>
```

6.3.7 Rule enhancements for selection lists (CONTAINS, SIZE, SELECT, DESELECT)

Input components

- CMS_INPUT_CHECKBOX
- CMS_INPUT_COMBOBOX
- CMS_INPUT_LIST and
- CMS_INPUT_RADIOBUTTON

allow users to choose from the values that have been defined by the project developer. Alternatively, the CMS_INCLUDE_OPTIONS data element can be used to populate them with values dynamically, e.g., with datasets from a referenced table, with project languages, or with presentation channels, etc. As a result, one or more entries (depending on the type) can then be selected from the input component or the selection can be left empty.

Previously, you could only use the rules to check whether or not these input components were empty (<EMPTY/> tag), but with the release of FirstSpirit version 5.2 it is now possible to check the following aspects for CMS_INPUT_CHECKBOX and CMS_INPUT_LIST as well:

- How many entries (SIZE property) have been selected
- Which entries (<CONTAINS/> tag) have been selected

As well as allowing you to check selected entries against text, the <CONTAINS/> tag can be used to check entries against the properties of other input components in the same form, e.g., against a selected entry in CMS_INPUT_RADIOBUTTON.

Furthermore, it is now also possible to manipulate the selection of all four input component types using the following properties:

- SELECT
- DESELECT

As with CMS_INPUT_COMBOBOX, query-based population via rules (VALUE property) is now also supported by CMS_INPUT_RADIOBUTTON, CMS_INPUT_CHECKBOX, and CMS_INPUT_LIST as well.

For more information, see the FirstSpirit Online Documentation,

- *"Template development / Rules / Comparative expressions / <CONTAINS/>"*
- *"Template development / Rules / Form properties <PROPERTY/> / Property SIZE"*
- *"Template development / Rules / Form properties <PROPERTY/> / Property DESELECT"*
- *"Template development / Rules / Form properties <PROPERTY/> / Property SELECT"*
- *"Template development / Rules / Form properties <PROPERTY/> / Property VALUE"*

6.3.8 Checking whether a section has been selected in FS_REFERENCE (SECTION)

The FS_REFERENCE input component can be used to incorporate any reference. When a page, page reference, or global page with at least one section is selected, a selection list appears so that a specific section can be selected. In this way, anchor links ("anchors") can be included on web pages.

You can use the SECTION property to check whether a section has been selected, e.g.:

```
<RULES>
    <RULE>
        <WITH>
            <NOT_ NULL>
                    <PROPERTY name="SECTION" source="st_link"/>
            </NOT_NULL>
        </WITH>
        <DO>
            <VALIDATION scope="RELEASE">
```

```
                <PROPERTY name="VALID" source="st_link"/>
                <MESSAGE lang="DE" text="Sie können zusätzlich
                      einen Absatz auswählen"/>
                <MESSAGE lang="*" text="You can select a section
                      additionally"/>
            </VALIDATION>
        </DO>
    </RULE>
</RULES>
```

In this example, a check is being performed to see if a section has been selected in an FS_REFERENCE component ("st_link" identifier). If it has not, the message "You can select a section additionally" appears.

For more information, see the FirstSpirit Online Documentation, "Template development / Rules / Form properties <PROPERTY/> / Property SECTION".

6.3.9 Code completion for rules

In order to support template developers better during rule creation, with version 5.2, a code completion feature has been introduced on the "Rules" tab. At the press of a button, you can use this code completion feature to show all the tags and corresponding parameters (plus values) that are available for the rule definition in the current syntactic context, and can insert them at the insert mark on the Rules tab, e.g.,:

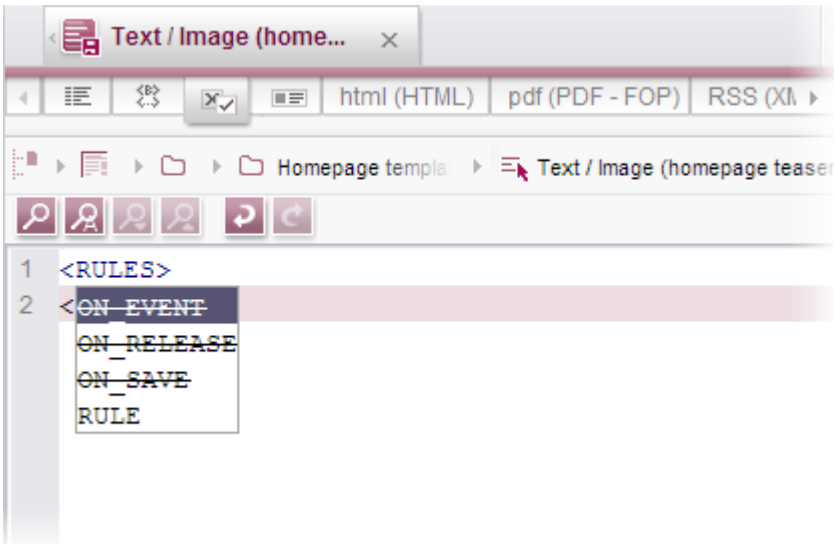


Figure 6-3: Auto-completion on the "Rules" tab

For this purpose, the insert mark must be positioned specifically within the code. For more information, see also the FirstSpirit Online Documentation, chapter "Interesting

facts / Input tools / Rules tab".

You can look up tags and parameters for rules and find out about their meaning and syntax by referring to the FirstSpirit Online Documentation, chapter "Template development" / "Rules".

Outside of forms, other entries in the SiteArchitect workspace can now also be checked in FirstSpirit version 5.2. See chapter 6.4, page 79 for more information.

6.4 Validation outside of forms

As of FirstSpirit version 5.2, checks (via "rules") are no longer restricted to entries in forms. Even entries located outside of forms in the SiteArchitect workspace are now checked by default and any invalid entries are shown in a uniform manner, e.g.:

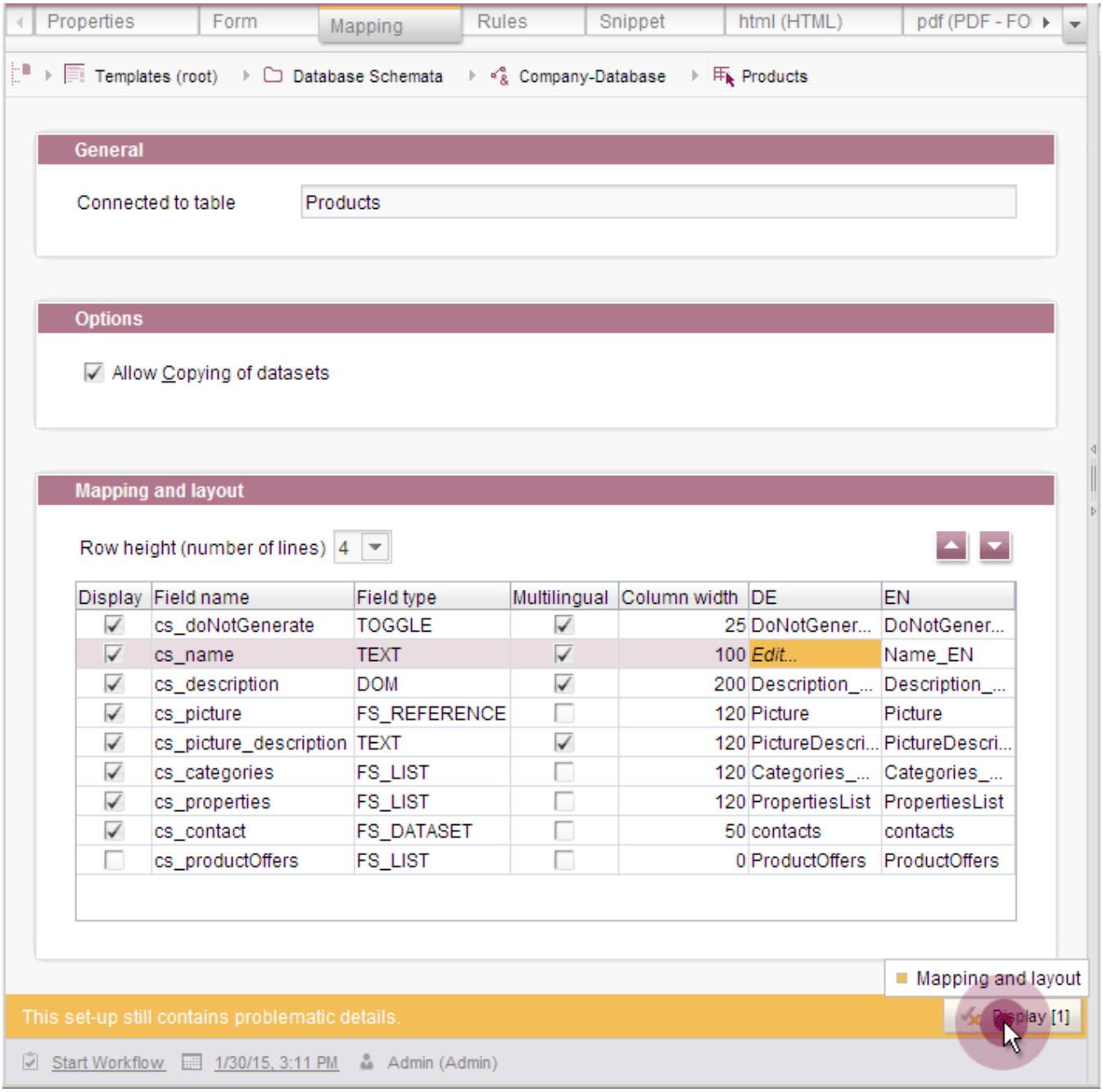


Figure 6-4: Table template – "Mapping" tab

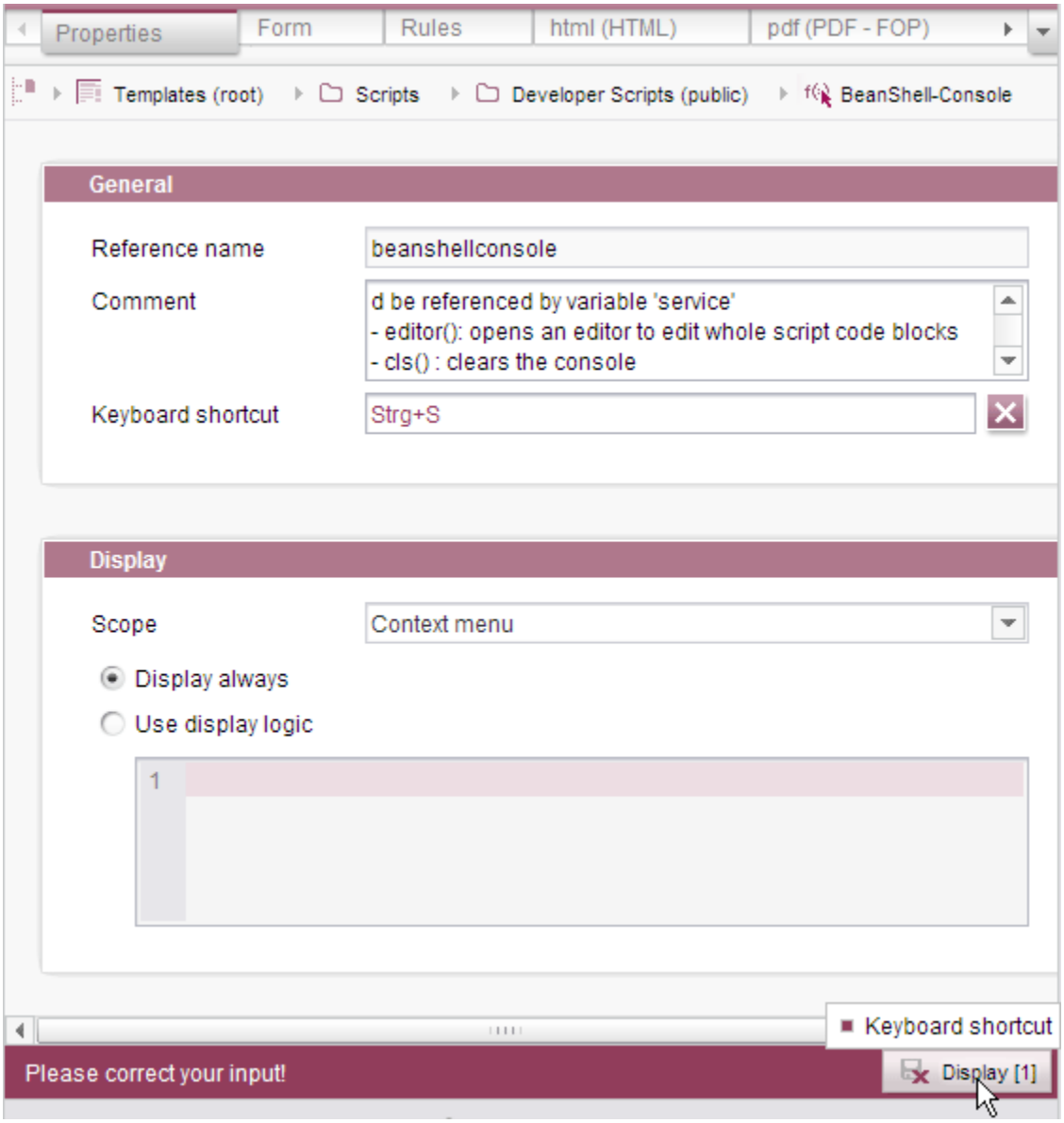


Figure 6-5: Scripts and workflows – "Properties" tab

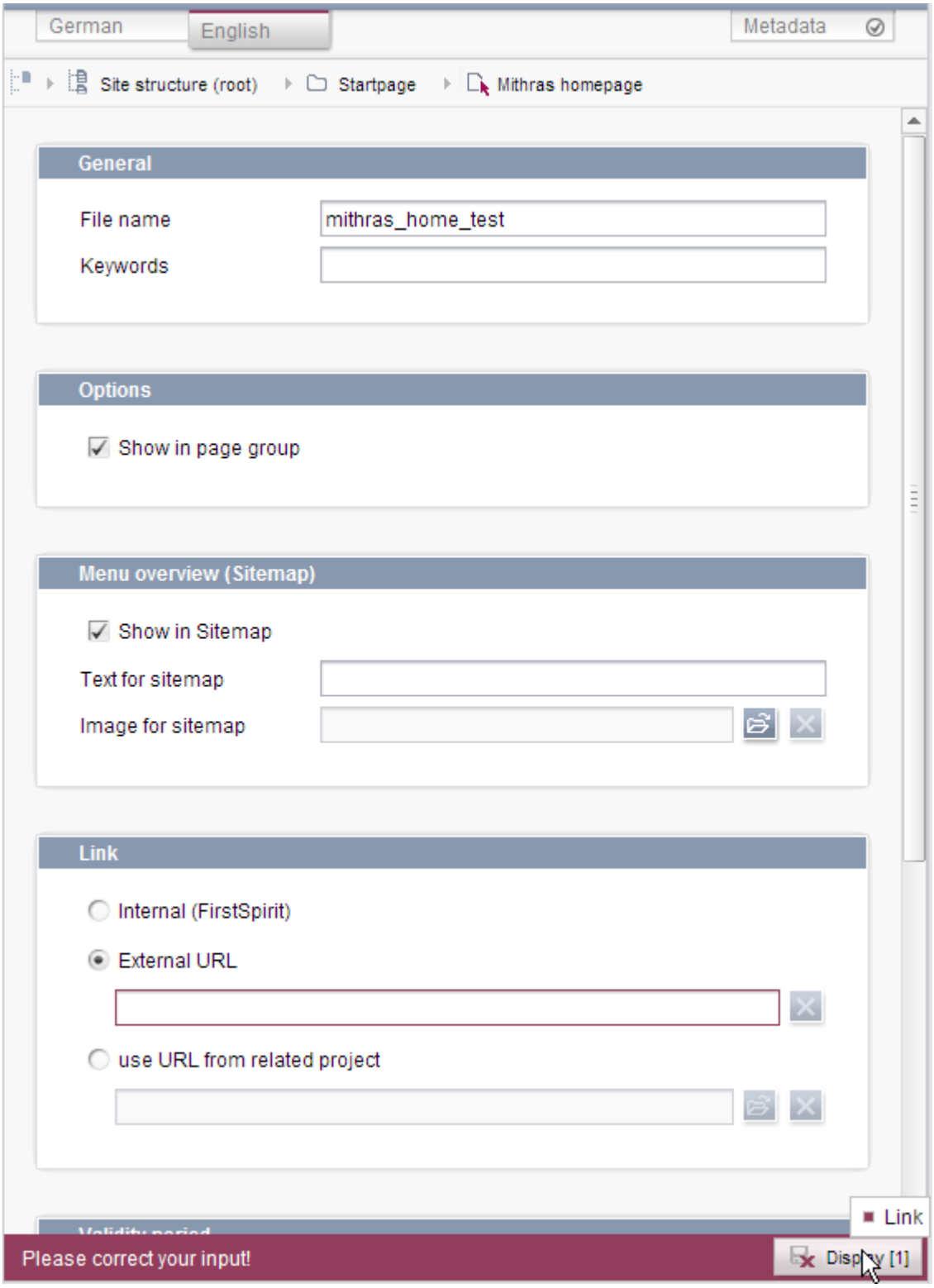


Figure 6-6: Page reference

6.5 FS_DATASET: New selection mode

Where datasets in FS_DATASET in SiteArchitect were previously selected via a selection dialog, they can now be selected from a simple list. This can have a positive effect on performance. See also chapter 5.2.4, page 56 for more information.

`selectorMode=list` must be set for this purpose.

Example:

```
<FS_DATASET
        name="st_contact"
        allowDelete="no"
        allowEdit="no"
        allowNew="no"
        hFill="yes"
        selectorMode="list"
        useLanguages="no">
        <LANGINFOS>
          <LANGINFO lang="*" label="Contact" />
          <LANGINFO lang="DE" label="Kontakt" />
        </LANGINFOS>
        <SOURCES>
          <CONTENT name="contacts"/>
        </SOURCES>
</FS_DATASET>
```

Prerequisite: Only one data source is defined with the SOURCES / CONTENT tags. If SOURCES / CONTENT is not specified or if more than one data source is defined, the selection dialog (`selectorMode=dialog`) is used automatically.

If the "combobox" value is set for the `mode` parameter (`mode="combobox"`), the `selectorMode` parameter is rendered ineffective.

This parameter does not affect either the display or the selector mode in ContentCreator.

6.6 External synchronization of project properties

With the release of FirstSpirit version 5.2, the "external synchronization" functionality not only allows you to import/export project content that has been entered using SiteArchitect and/or ContentCreator, but also to import/export (synchronize) project properties as well, including across servers. This method can be used, for example, to transfer the properties of a project to an empty project as well as to synchronize the project configuration of multiple projects.

Project properties can be synchronized via the software interface under "Project properties" in the "External synchronization" area in SiteArchitect (see Figure 6-7, see also chapter 6.6.1, page 84) or using the FirstSpirit Developer API, `de.espirit.firstspirit.store.access.nexport.operations` package.

The permissions of a project administrator are a minimum requirement in order to import and export project properties; server administrator permissions are required to import server properties (see the list in chapter 6.6.1).

In principle, the data format used for this special export function when FirstSpirit version 5.1 is initially released is to be retained in subsequent (minor or release) versions. However, given the highly complex structure involved, the first step is to gather empirical data regarding this new function from real projects (including from customers and partners). Based on this empirical data, later versions may see a change in the data format in which even the introduction of minor changes may be incompatible. User feedback is expressly desired in this case to make it possible to respond to partner requests.

For more information, see the *FirstSpirit Online Documentation,* "Further topics / External synchronization". For more information about the project and server properties that are synchronized, see the *FirstSpirit Documentation for Administrators*, "Server properties" and "Project properties" chapters.

6.6.1 External synchronization of project properties in SiteArchitect

Some project properties are referenced to server properties, e.g., users, groups, schedules, etc. You can choose whether these server properties should also be included in a synchronization run.

Whether project properties (and referenced server properties) should also be synchronized can be selected under "Project properties" in the "External synchronization" area of SiteArchitect. Click "Inactive" to open a dialog, where you can select some (or all) of the properties as required:

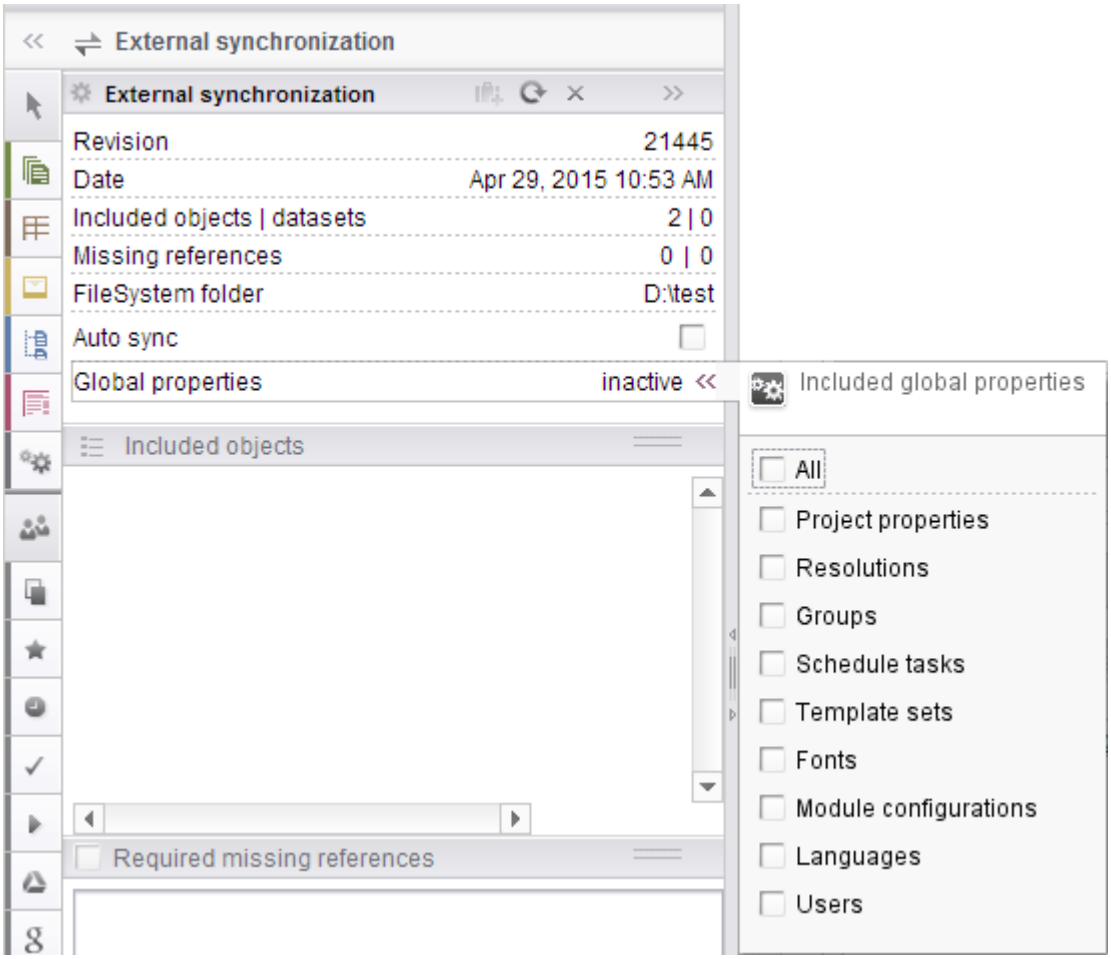


Figure 6-7: External synchronization – Project properties

For more information, see the *FirstSpirit Online Documentation*, "Further topics / External synchronization / Project properties".

6.7 Multi Perspective Preview

As Internet-enabled mobile devices such as notebooks, tablet PCs, and smartphones become more and more widespread, website designs need to be more and more flexible, with content which can be displayed perfectly on different display geometries and in different resolutions. That is why FirstSpirit version 5.1 introduced an easy way

for editors to check what website content looks like and how well it can be navigated with a variety of display sizes in ContentCreator, while also allowing content, layouts, and images to be perfectly adapted for the output device concerned. Along with size considerations, other aspects can also be taken into account, e.g., previews for specific user groups ("Multi Perspective Preview", "MPP").

With the release of FirstSpirit version 5.2, the Multi Perspective Preview concept has been carried across to SiteArchitect as well. To enable Multi Perspective Preview mode in SiteArchitect, go to the "MPP" tab in the integrated preview. It is operated in exactly the same way as in ContentCreator.

6.7.1 Accessing time adjustment and values specified by editors (JavaScript)

In FirstSpirit version 5.1, JavaScript object `WE_API` was used to query the current time setting plus the values the editor had specified in the MPP using JavaScript. So that the Multi Perspective Preview can be used in SiteArchitect as well, API object `MPP_API` (see `MPPWebControl.IDENTIFIER`) has now been made available in JavaScript as of FirstSpirit version 5.2. It is defined by the `de.espirit.firstspirit.client.mpp.MPPWebControl` interface and can now be used in both clients. The previous mechanisms that were implemented on the basis of JavaScript object `WE_API` have been deprecated in FirstSpirit version 5.2 and should no longer be used (see also FirstSpirit Developer API, `de.espirit.firstspirit.webedit.client.api` package, `Preview`interface).

Example of how to query the currently selected time
FirstSpirit version 5.1:

```
WE_API.Preview.getTimeParameter()
```

FirstSpirit version 5.2:

```
MPP_API.getTimeParameter()
```

Example of how to query a value specified by the editor
FirstSpirit version 5.1:

```
WE_API.Preview.getParameter("role");
```

Here, `role` is the variable name of the input component that the editor uses to select a specific role, such as that of a site visitor.

FirstSpirit version 5.2:

```
MPP_API.getParameter("role");
```

For more information, see the FirstSpirit Online Documentation, "Advanced topics / Multi Perspective Preview / JavaScript API".

6.7.2 Accessing time adjustment and values specified by editors (JSP)

In FirstSpirit version 5.1, the current time setting and the values specified by the editor in MPP using JSP were queried as follows: The selected values were written directly to the HttpSession interface while also using the `fs.preview.` namespace. As part of FirstSpirit version 5.2, `de.espirit.firstspirit.client.mpp.PreviewParameter` has been introduced to serve as an encapsulating object. This can be found under the "$fsmpp" HttpSession attribute (see `de.espirit.firstspirit.client.mpp.PreviewParameter.SESSION_ATTRIBUTE`)

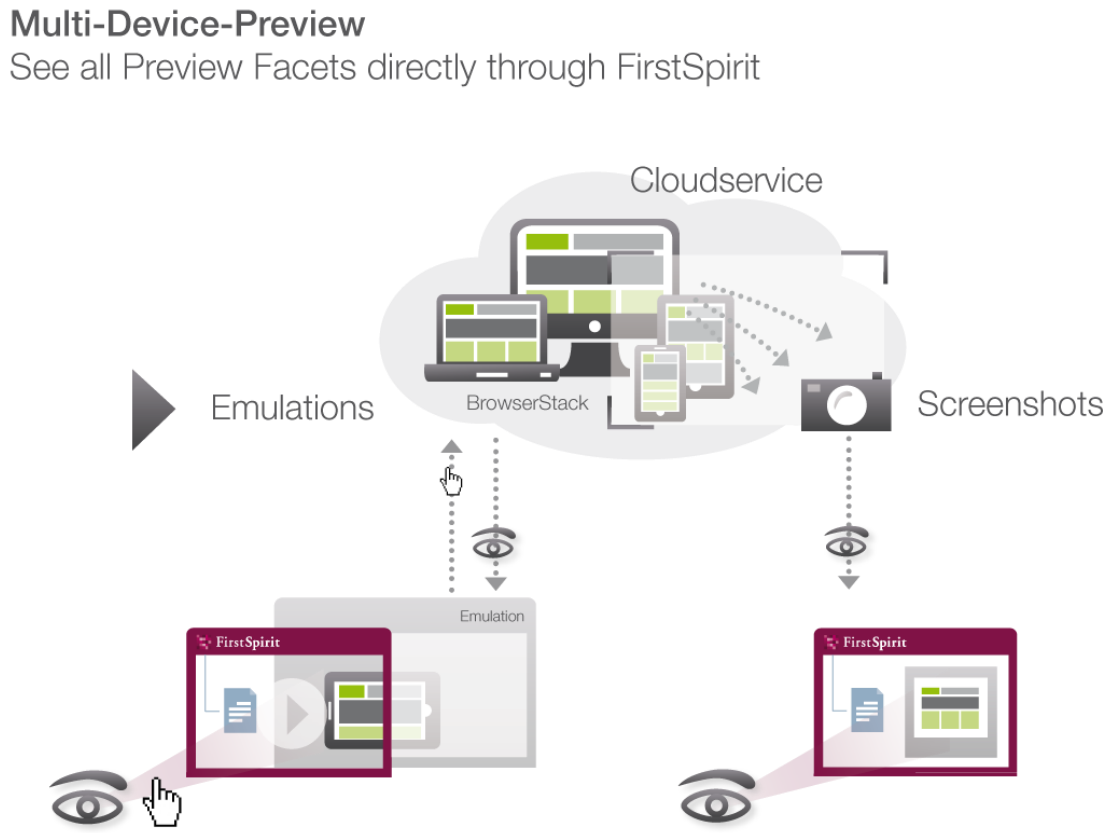
Example:

```
<%
  de.espirit.firstspirit.client.mpp.PreviewParameter
  mpp = PreviewParameter.Factory.from(request);
  if (mpp != null)

  {
     %>MPP Time: <%= mpp.getTimeParameter() %><%
  }

%>
```

6.8 Extending preview capabilities and harnessing external options

FirstSpirit version 5.2 introduces FirstSpirit version 5.2 introduces another way of testing content and layouts in various display sizes and for a variety of device and browser types by making use of the FirstSpirit Developer API. The `de.espirit.firstspirit.client.plugin.ExternalPreviewItemsPlugin` interface has been specially created for this purpose. It allows the preview icon in the horizontal tool bar to be extended so that website testing services (for example) can be integrated in accordance with the best-of-breed philosophy. Within this context, `ExternalPreviewItems` can be used to provide URLs that each represent a particular incarnation of the preview (see also chapter 5.2.2, page 53 and onward). An implementation example incorporating an external "BrowserStack" web service (http://www.browserstack.com) has been created. Please note that there is a charge for

using this service! The implementation example allows you to see what the content looks like in the AppCenter area with a wide range of browsers and device types, as well as in various versions and with different operating systems (e.g., Windows, Mac OS, Android, etc.).. In this case, the display is not simulated by resizing the original display geometry (as with the "MPP", chapter 6.7, page 85). Rather, the display is based on a real browser/system environment. Not only does this mean that the site can be displayed under highly realistic conditions but it also means that project developers benefit from excellent debugging facilities.



As part of this feature, the `de.espirit.firstspirit.agency.GenerateElementOperation` interface has also been created. You can use this interface to trigger a (synchronous) generation process (potentially with deployment) that returns the URL via which the page can be accessed.

For an example implementation, see the *FirstSpirit online documentation, "Plug-In Development / Examples"*.

6.9 Chrome Developer Tools

With its "DevTools", Google Chrome provides developers with an important set of tools for implementing web projects.

If "Google Chrome" has been configured as the browser engine in SiteArchitect (menu: "View / Browser Engine"), FirstSpirit version 5.2 and higher allows you to access these tools in the AppCenter area / via the context menu in the integrated preview by selecting the "Developer tools" entry:

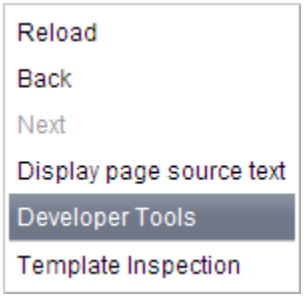


Figure 6-8: Context menu in the integrated preview

A new window opens with the developer tools:

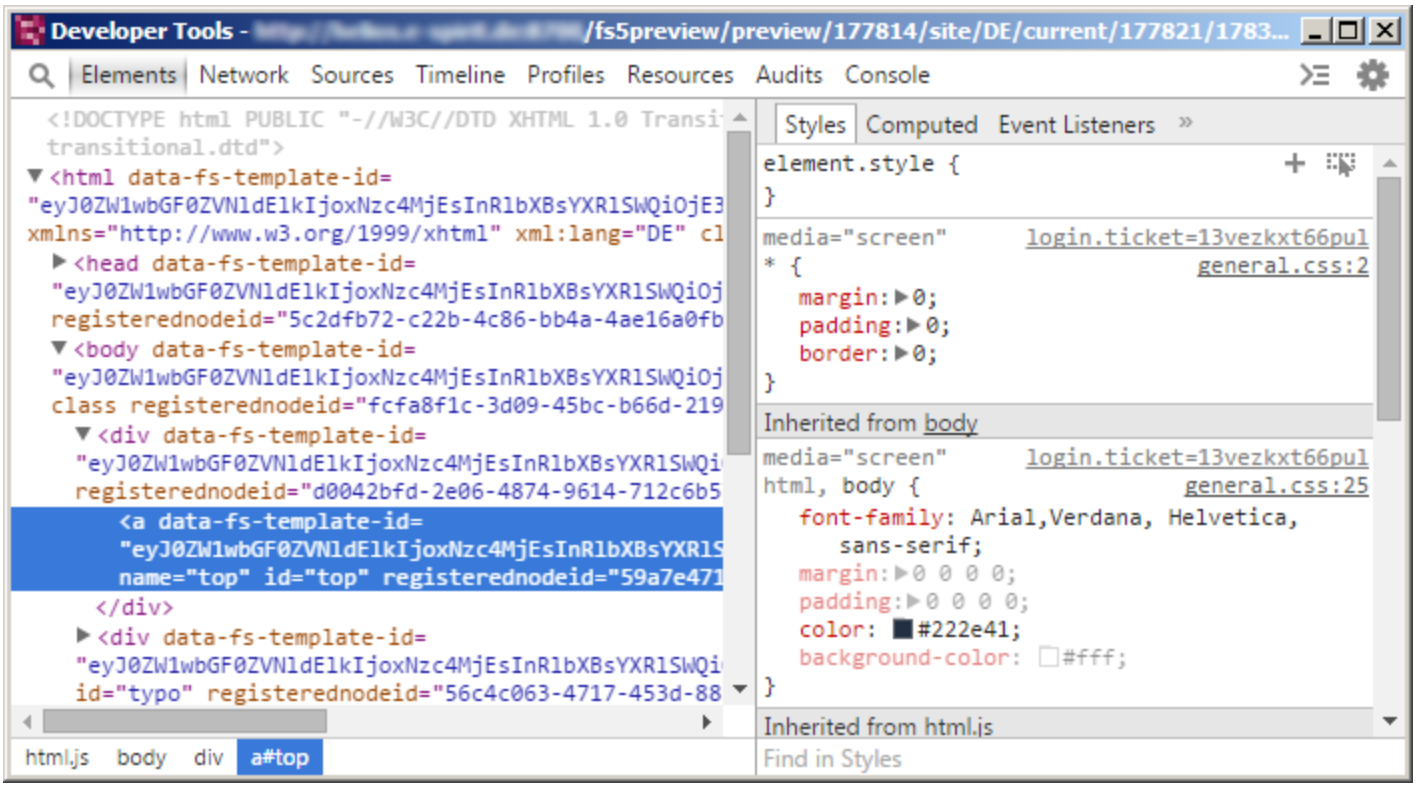


Figure 6-9: Chrome Developer Tools

6.10 Enhancements in the "Snippet" area

In addition to the GUI changes described below, the FirstSpirit Access API has been enhanced in relation to snippets. See chapter 6.13.2, page 114 for more information.

6.10.1 Outputting dataset IDs in snippets

At numerous points, elements known as "snippets" can be used to control how FirstSpirit objects are displayed in SiteArchitect and ContentCreator.

With the release of FirstSpirit version 5.2, it is now possible to output the ID of datasets, both for "labels" and for "extracts". This is achieved by means of:

```
#id
```

Within the relevant table template, a snippet definition for press releases, including the relevant dataset ID, might look like this:

Label:

```
"Press release " + italic(#id)
```

Example of what is displayed in SiteArchitect:

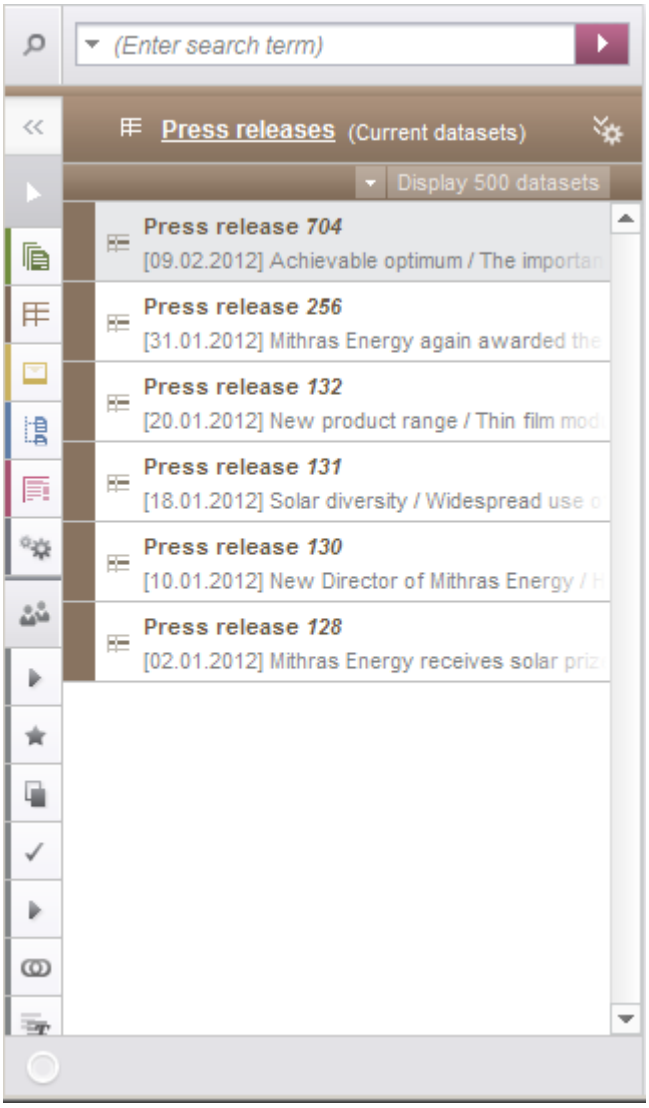


Figure 6-10: Dataset ID in the brief overview of data sources

Example of what is displayed in ContentCreator:

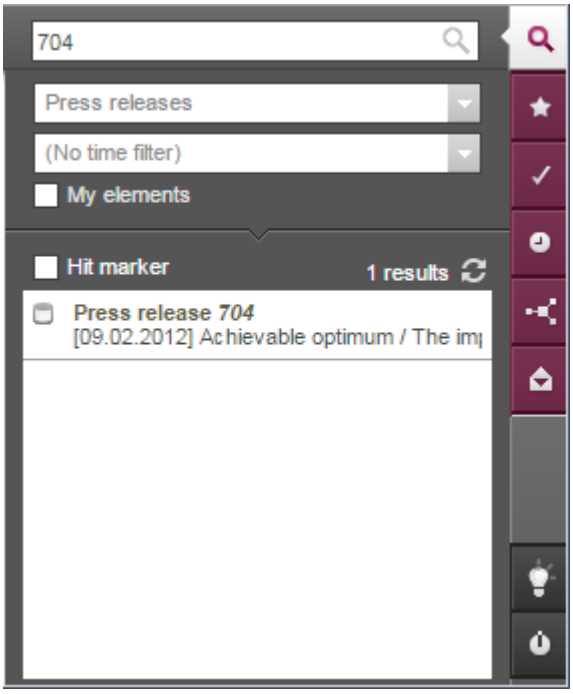


Figure 6-11: Dataset ID in search function

For more information on working with snippets, see the *FirstSpirit Online Documentation*, "Template development / Snippets".

6.10.2 Previewing link template snippets

Since the release of FirstSpirit version 5.1R1, it has also been possible to define snippets for link templates (Template store, Link templates, "Snippets" tab). These are then used for tooltips in ContentCreator. For more information, see also the *FirstSpirit ContentCreator documentation*, "Editing a preview page / Input elements / Rich text editor", section "Insert/edit link".

With the release of FirstSpirit version 5.2, the preview function for testing snippet definitions in SiteArchitect as early as the template development stage has now been made available for link templates as well.

The relevant data is entered in the dialog that can be accessed by pressing the "Preview data" button on the "Properties" tab:

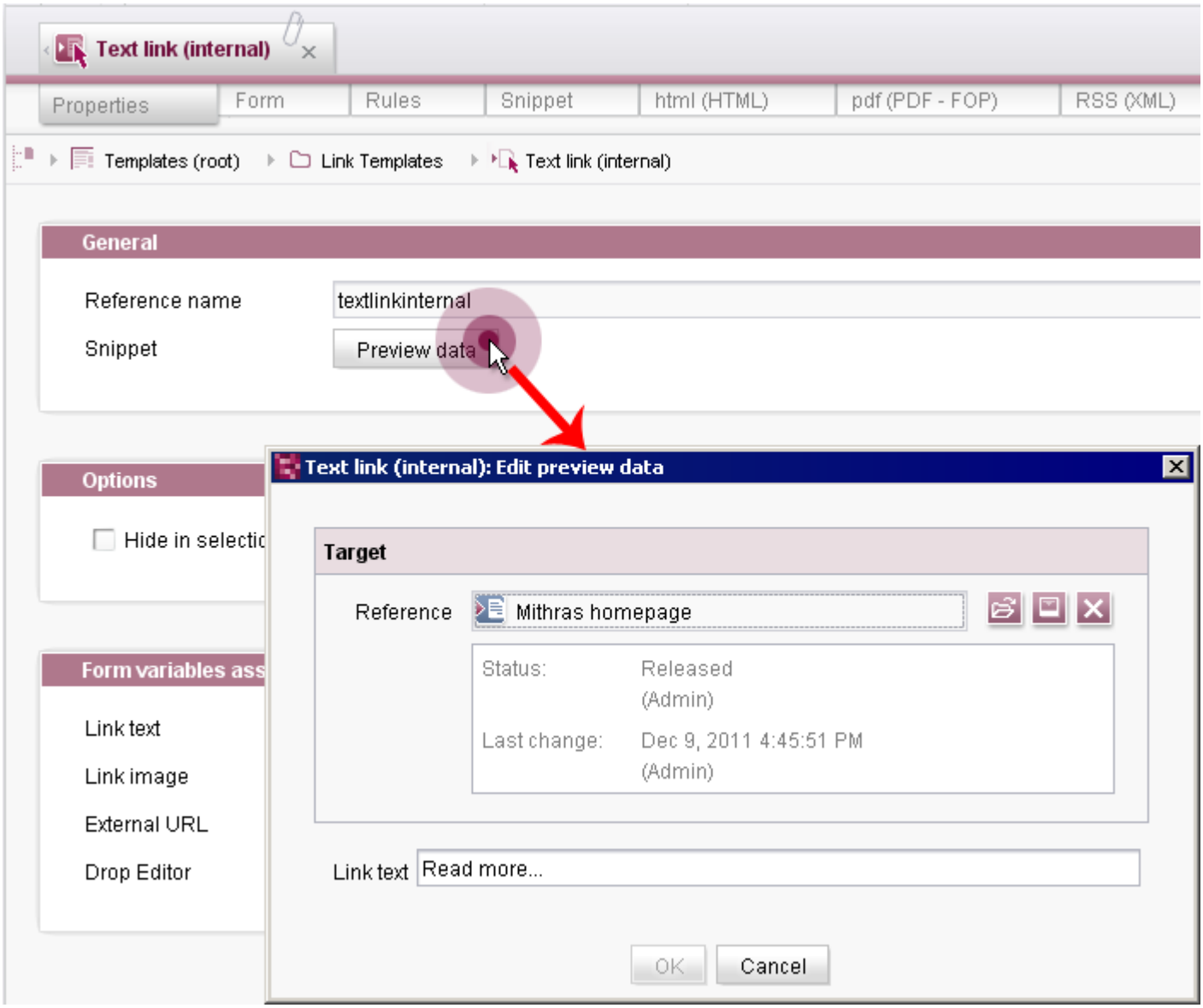


Figure 6-12: Configuring the preview function for snippets in link templates

When used in conjunction with a suitable definition on the Snippets tab within the workspace, the preview is then displayed in the AppCenter area of the link template ("Snippets" tab). This means that content from

- input components of the associated template ("Form" tab) or
- objects referenced in the input components of the template (e.g., pages referenced using FS_REFERENCE, page references, media, or datasets referenced using FS_DATASET)

can be used.

For more information, see also the *FirstSpirit Online Documentation*, "Template development / Snippets", section "Snippets in link templates".

6.11 Enhancements linked to generation

6.11.1 Advanced URL Creator

For FirstSpirit 5.0, an API interface was developed that can be used to create custom "URL generators" (in the form of modules), which can in turn generate URLs on demand. This makes it possible to undertake search machine optimization (SEO) by using "talking" URLs that are easier for users to understand and are easier for search machines to analyze, for example.

A reference implementation ("Advanced URL Creator") has been made available to supplement the new interface. It supports the integration of a variety of path strategies for generating URLs in FirstSpirit. The reference implementation can be selected via the "Advanced URLs" setting in the project properties. In this context, the generation of advanced URLs is being delegated to the UrlFactory factory.

Regardless of whether the reference implementation ("Advanced URL Creator") or a separate customer-specific path strategy based on the new interfaces is implemented and integrated into FirstSpirit as a module, the following applies:

In the case of self-referencing (i.e., the page currently being generated includes a reference to itself, e.g., in the navigation), an empty string " " is returned as the URL (default behavior since FirstSpirit version 5.0). A URL of this type that conforms to RFC is interpreted as a link to the current page.

However, not all browsers are able to work with these "empty" links. Cases have been known in which the browser interprets an empty link not as a link to the URL that is currently being displayed but as a link to the directory path on which the generated file is located. Instead of reloading "/index.html", the browser attempts to open the current directory "/". In cases like this, a reference that is output in the HTML presentation channel with the expression `$CMS_REF(..)$` in a `<a href>` tag will lead to the affected pages not being reloaded. This poses problems for dynamic web pages in particular. In combination with the `useWelcomeFilenames=false` parameter, this behavior would ultimately lead to an http 404 error due to the failure to locate an index

file

To avoid these browser-specific problems, a new configuration parameter – selfLink – is being introduced with FirstSpirit version 5.2 and the existing standard behavior is being modified.

For more information, see the *FirstSpirit Online Documentation*, "Advanced topics / Generation / Advanced URLs / Configuration", "selfLink" subsection.

Enhancements and changes have also been made to the API in this area. See chapter 6.13.3.3, page 117 for more information.

6.11.2 Delta generation

Delta generation describes the generation of content which has actually changed since it was last generated.

In the case of a content projection (output of datasets via integration of a required data source on a page (Page Store), configuration is via the corresponding page reference) in earlier versions of FirstSpirit, if a delta generation was performed further to changing and releasing a dataset, only details pages on which the changed dataset was displayed were regenerated (Page reference / "Content" tab / "Number of entries per page": 1). This behavior has changed in FirstSpirit 5.2: Now, pages on which the changed dataset is displayed and for which the setting for "Number of entries per page" on the "Content" tab for a page reference is 0 or greater than 1 ("Overview page") are also regenerated.

The behavior described above can be modified further with the new parameter introduced for the

```
UPDATE_PAGEREF_WHEN_ENTITY_HAS_CHANGED
```

DependencyRules:

If `UPDATE_PAGEREF_WHEN_ENTITY_HAS_CHANGED` is **set** (or if no DependencyRules are set), all details pages and the overview page are included in a delta generation.

If `UPDATE_PAGEREF_WHEN_ENTITY_HAS_CHANGED` is explicitly **not set**, only the details page concerned and the overview page are included in a delta generation.

For more information about delta generation, see also the

- FirstSpirit Online Documentation, "Advanced topics / Generation".
- FirstSpirit Community
- FirstSpirit Developer API:
    - `DeploymentUtil`
    - `DeltaGeneration`
    - `DeltaGeneration.ChangeSet`

  (all in the `de.espirit.firstspirit.access.schedule` package)

6.12 Changes and enhancements in relation to ContentCreator

6.12.1 Multi-tab support

In earlier versions of FirstSpirit, editors could only have one ContentCreator project open in each instance of the browser. This meant, for example, that a link within a workflow notification e-mail could not be followed if ContentCreator had already been opened with another project.

As of FirstSpirit version 5.2, it is now possible to open multiple projects within the same instance of the browser. As a result, there are several possible scenarios:

- 1 editor, n instances of the project: The editor can open the same project on multiple browser tabs at the same time.
- 1 editor, n projects: The editor can open various different projects at the same time.

For more information, see also chapter 5.1.1, page 28.

From a technical perspective, this functionality has been implemented by means of HttpSubSessions. By default, HttpSubSessions are activated via a corresponding filter in the web.xml file.

With servlet API 2.4 and higher, a different default setting applies and so filters are only activated in the event of direct requests and not when they are of the "Forward" type. In this case, the web.xml file (of the web app component) must be adapted manually:

```
<filter-mapping>
          <filter-name>WebAuthentication</filter-name>
          <url-pattern>/</url-pattern>
          <dispatcher>FORWARD</dispatcher>
</filter-mapping>
```

If the relevant web app component is not just used in ContentCreator, multiple dispatchers can – of course – also be defined:

```
<filter-mapping>
          <filter-name>WebAuthentication</filter-name>
          <url-pattern>/</url-pattern>
          <dispatcher>REQUEST</dispatcher>
          <dispatcher>FORWARD</dispatcher>
</filter-mapping>
```

For more information, see the *FirstSpirit Manual for Administrators*.

6.12.2 LiveEdit mode

As of FirstSpirit version 5.2, it is now possible to edit content directly on a page that has already been deployed (e.g., an intranet page). For this purpose, corresponding edit icons now appear so that editors can change text very quickly and more directly, or even upload images. Naturally, all the permissions of the editors concerned are evaluated during this process and, if necessary, the changes can be released by workflow.

For more information, see also chapter 5.2, page 26.

For direct and smooth use of LiveEdit mode, the user should be logged into the FirstSpirit server.

To enable use of LiveEdit mode, JavaScript code must be inserted into the required presentation channel for the respective page template. The LiveEdit icons are shown in the same way as the EasyEdit icons and on the basis of the same "editorIds".

Before changes can be released and deployed, a corresponding workflow must be present in FirstSpirit ServerManager:

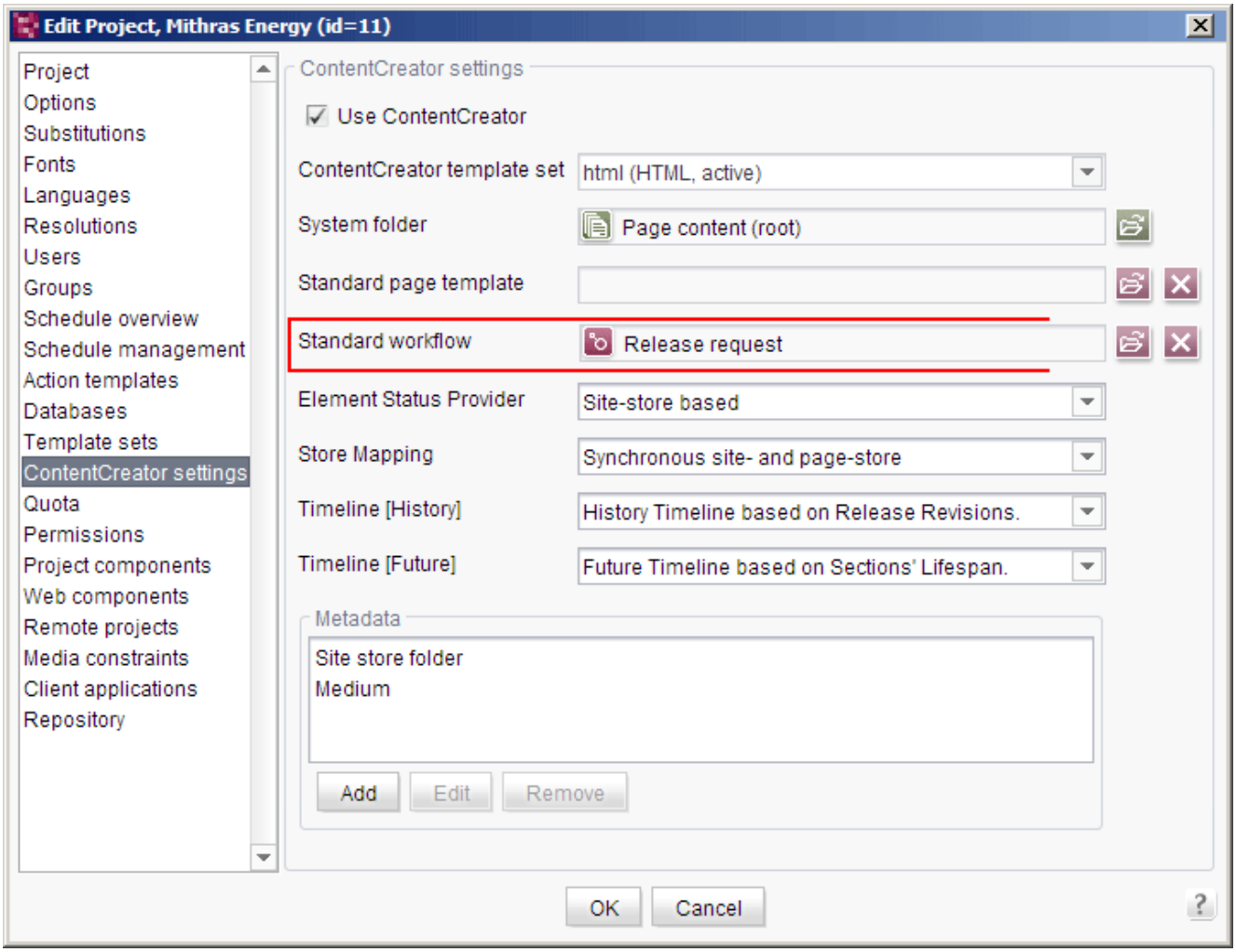


Figure 6-13: Defining the workflow for LiveEdit

Alternatively, you can use the *workflow* parameter in the relevant page template to specify the ID of the workflow that is to be executed on saving, e.g.:

```
<script
     src="http://SERVER:PORT/fs5webedit/live/api.jsp?project=XYZ
     %workflow=ID" ...>
```

Here, *ID* is the ID of the project workflow required (select Alt + P on the workflow node or the "Extras – Display properties" context menu).

Once the changes to the pages, sections, or datasets have been saved (with the "Save" button), another window for starting the workflow appears. As a result, the modified content can be released and deployed.

If project-specific plug-in developments are used (gadgets and others), it is necessary to utilize an instance of ContentCreator that is local to the project. For more information, see also the FirstSpirit Manual for Administrators, chapter "ContentCreator as a local project application".

For more information, see

- FirstSpirit Online Documentation, "Template development / ContentCreator / LiveEdit"
- FirstSpirit Manual for Administrators, chapter "ContentCreator settings"

6.12.3 Using drag-and-drop to create content

The ability to create links in ContentCreator by using drag-and-drop in the DOM editor was implemented back in FirstSpirit version 5.1. In version 5.2, this option has been extended to page content, which means that you can now also use drag-and-drop to create pages, sections, and datasets. For more information, see also chapter 5.1.4, page 33. For FS_BUTTON handler classes

- `de.espirit.firstspirit.webedit.server.executables.NewSectionExecutable`
- `de.espirit.firstspirit.webedit.server.executables.NewPageExecutable`
- `de.espirit.firstspirit.webedit.server.executables.NewContentExecutable`

support has, therefore, been implemented so that they can be used in `onDrop` attributes of FS_BUTTON components.

As of FirstSpirit version 5.2, a blackberry-colored frame is applied to FS_BUTTON instances in the ContentCreator preview (just like other elements) if the `onDrop` parameter has been defined and the user selects an element with the mouse in ContentCreator and then hovers over the FS_BUTTON with the button still pressed.

Note: When content is created using drag-and-drop, sections and datasets may be created directly. As a result, they may be displayed on the preview page without the editor having an opportunity to enter content via form windows. If the elements created using drag-and-drop do not have any content, it may be that nothing is displayed in the preview. For this reason, placeholder information should be provided so that it is displayed on the preview page and can be edited subsequently by the editor. Alternatively, rules or mandatory parameters (e.g., `allowEmpty="no"`) can be used

instead: If certain rules to prevent saving (`<VALIDATION scope="SAVE">`) are violated when content is created using drag-and-drop or if mandatory fields are left empty, the form window opens during the drag-and-drop operation so that the editor can enter content in accordance with the specified rules.

For more information, see also the FirstSpirit Online Documentation, chapter "Template development / Rules" and page "Template development / Forms / Default values".

The `DROPTYPES` tag of the FS_BUTTON input component can be used to define or restrict the object types that can be dragged onto the input component with the mouse pointer.

For more information, see also the FirstSpirit Online Documentation, page "Template development / Forms / BUTTON".

6.12.3.1 Pages

Pages can be created using drag-and-drop as follows:

- Via `editorId()` a **Navigation function** with the call (for more information, see the *FirstSpirit Online Documentation*, page "Template development / Content Highlighting and EasyEdit / Use in a project", section "Moving menu items in the ContentCreator preview")
- Via an appropriately configured **FS_BUTTON** input component, e.g.:
  (FS_BUTTON form in a page template)

```
<FS_BUTTON
    name="pt_createPage"
    hidden="yes"
    icon="fs:new"
    onClick="class:NewPage"
    onDrop="class:NewPage"
    style="firstspirit"
    useLanguages="no">
    <LANGINFOS>
      <LANGINFO lang="*" label="Create page"/>
      <LANGINFO lang="DE" label="Seite anlegen"/>
    </LANGINFOS>
  </FS_BUTTON>
```

(Output in HTML channel of page template)

```
...
$CMS_SET(isWebEdit, #global.is("WEBEDIT"))$
...
   $CMS_VALUE(#global.page.body("Content center"))$
          $CMS_IF(isWebEdit)$

                  <span$CMS_VALUE(
                        fsbutton(
                        editorName:"pt_createPage")
              )$></span>
          $CMS_END_IF$
 ...
```

For more information, see also the following pages of the *FirstSpirit Online Documentation*:

- "Template development / Forms / Input components / BUTTON"
- "Template development / ContentCreator / Functional scope", section "FS_BUTTON handler classes"
- "Template development / Template syntax / Functions / in instructions / fsbutton"

For information on the editor view, see also chapter 5.1.4.1, page 34.

6.12.3.2 Sections

Sections can be created using drag-and-drop as follows:

- Via an appropriately configured **FS_BUTTON** input component, e.g.:
  (FS_BUTTON form in a page template)

```
<FS_BUTTON
    name="pt_createSection"
    hidden="yes"
    icon="fs:new"
    onClick="class:NewSection"
    onDrop="class:NewSection"
    style="firstspirit"
    useLanguages="no">
    <LANGINFOS>
      <LANGINFO lang="*" label="Create section"/>
      <LANGINFO lang="DE" label="Absatz anlegen"/>
    </LANGINFOS>
  </FS_BUTTON>
```

(Output in HTML channel of page template)

```
...
$CMS_SET(isWebEdit, #global.is("WEBEDIT"))$
...
   $CMS_VALUE(#global.page.body("Content center"))$
          $CMS_IF(isWebEdit)$

                   <span$CMS_VALUE(
                         fsbutton(
                         editorName:"pt_createSection",
                                parameter:{
                                      "page":#global.page.id,
                                      "body":"Content center"
                                 }
                                   )$></span>
          $CMS_END_IF$
 ...
```

For more information, see also the following pages of the *FirstSpirit Online Documentation*:

- "Template development / Forms / Input components / BUTTON"
- "Template development / ContentCreator / Functional scope", section "FS_BUTTON handler classes"
- "Template development / Template syntax / Functions / in instructions / fsbutton"

If the object being dropped is

- a section or
- a section master copy,

a direct copy of the section is created.

If the object being dropped is

- a medium,
- a page,
- a page reference,
- a dataset, or
- text (from word processing programs, from web pages, etc.),

the object can be pre-assigned to the section that is to be created once it is dropped and can be used by the editor. The **Drop Editor** combo box in the relevant section template can be used to select which input component is to have the dropped object pre-

assigned to it, and the object will be saved to this input component accordingly.

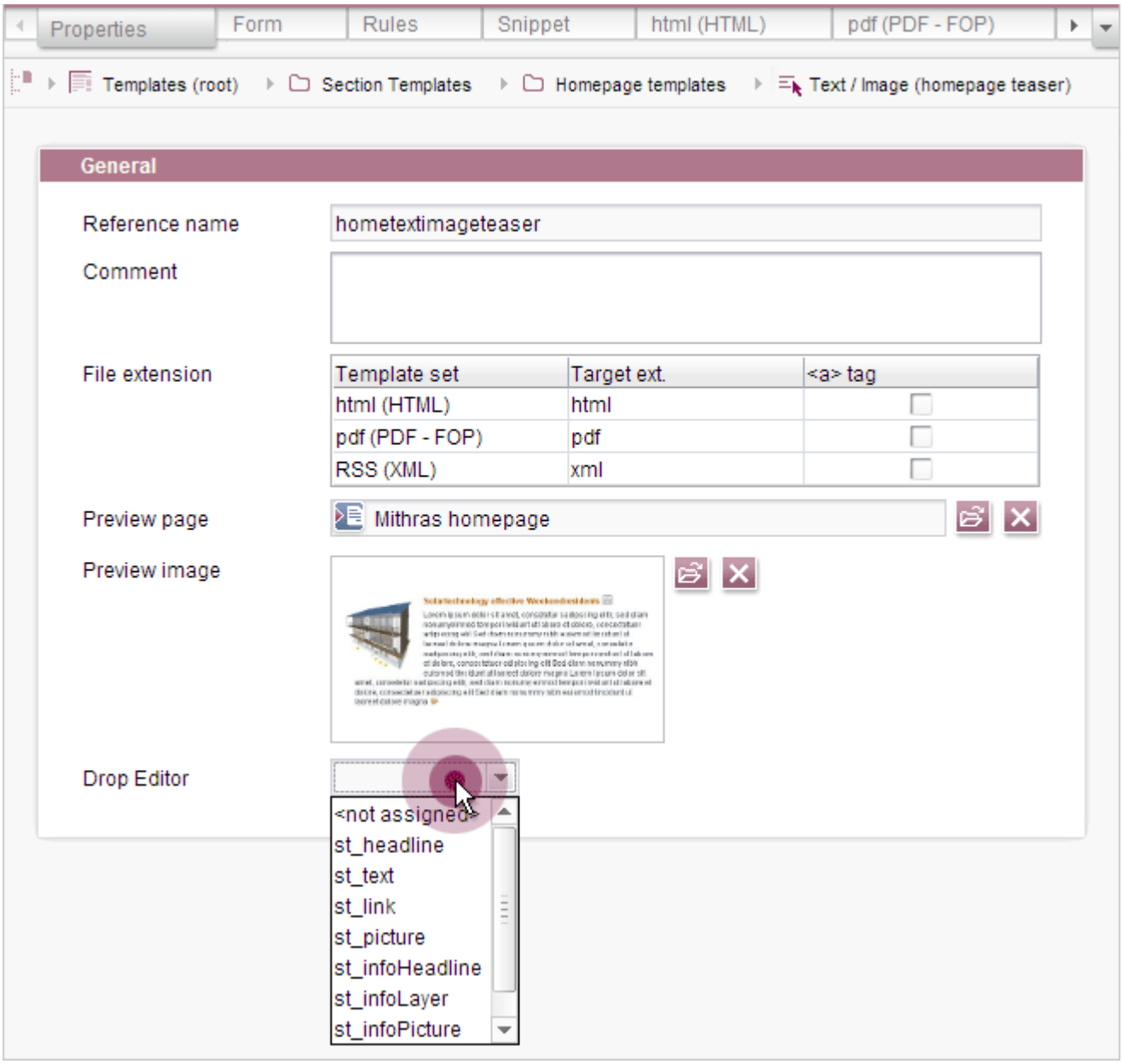


Figure 6-14: Section template – "Properties" tab – Drop Editor

This combo box shows all the identifiers for input components of the following types that have been defined on the "Form" tab:

- FS_BUTTON
- FS_REFERENCE
- FS_LIST

- FS_DATASET
- CMS_INPUT_IMAGEMAP
- CMS_INPUT_DOM
- CMS_INPUT_TEXT
- CMS_INPUT_TEXTAREA

are displayed here.

For example, FS_REFERENCE can be used to incorporate references on pages, page references, and media while FS_DATASET can be used to incorporate datasets. FS_BUTTON can be used in various scenarios, such as when you want to check which type of object may be dropped onto the drop zone (*DROPTYPES* / *MIME*, and *TYPE* tags), e.g., in the case of data originating from customer-specific reports. CMS_INPUT_DOM and other text input components can be used to incorporate dropped text.

In the context of ContentCreator, an editor who is performing a drop operation can choose from all section templates in which the input component type selected here is compatible with the object being dropped and which may be used in the relevant content area of the page (page template, tab: "Properties / Content areas / Allowed section templates"). If there is only one compatible section template, this is used automatically to create the section when the object is dropped (unless there are empty mandatory fields or certain rules are violated so as to prevent saving; for more information, see also the note in introductory chapter 6.12.3, page 99). During this process, the dropped object is saved in the input component that was selected under "Drop Editor".

If multiple section templates are available for selection, the editor can select the required template from a list. When the form opens, the object that is being dropped is pre-assigned to the input component selected under "Drop Editor". The editor can choose to accept it or – depending on how the input component is configured – can change/delete it, and enter and save any additional section content in the usual manner. If the section that is being created contains mandatory fields (e.g., because of rules that are designed to prevent saving or the `allowEmpty="no"` parameter), the form likewise opens.

If the object being dropped is a section that is compatible with the `body` parameter, a direct copy of the section is created.

If no corresponding input component has been defined on the Form tab, the selection remains empty (<not assigned>). If <not assigned> is selected, the section template is not used for a drop operation (exception: if a compatible section is dropped).

For information on the editor view, see also chapter 5.1.4.2, page 36.

6.12.3.3 Datasets

Datasets can be created using drag-and-drop as follows:

- Via an appropriately configured **FS_BUTTON** input component, e.g.:
  (FS_BUTTON form in a page template)

```
<FS_BUTTON
    name="pt_createDataset"
    hidden="yes"
    icon="fs:new"
    onClick="class:NewContent"
    onDrop="class:NewContent"
    style="firstspirit"
    useLanguages="no">
    <LANGINFOS>
      <LANGINFO lang="*" label="Create dataset"/>
      <LANGINFO lang="DE" label="Datensatz anlegen"/>
    </LANGINFOS>
  </FS_BUTTON>
```

(Output in HTML channel of page template)

```
...
$CMS_SET(isWebEdit, #global.is("WEBEDIT"))$
...
   $CMS_VALUE(#global.page.body("Content center"))$
          $CMS_IF(isWebEdit)$

                  <span$CMS_VALUE(
                        fsbutton(
                        editorName:"pt_createDataset",
                              parameter:{
                                    "content2":"products"
                               }
                                 ))$></span>
          $CMS_END_IF$
 ...
```

For more information, see also the following pages of the *FirstSpirit Online Documentation*:

- "Template development / Forms / Input components / BUTTON"
- "Template development / ContentCreator / Functional scope", section "FS_BUTTON handler classes"
- "Template development / Template syntax / Functions / in instructions / fsbutton"

The following settings must be applied:

1. Table template:
    - "Properties" tab / "Drop Editor": A suitable input component must be selected.
2. Page template:
    - "Form" tab: FS_BUTTON definition with `onDrop="class:NewContent"`
    - "HTML" tab: Output of FS_BUTTON via `fsbutton` method with "`content2`" parameter

If the object being dropped is

- a medium,
- a page,
- a page reference,
- a dataset, or
- text (from word processing programs, from web pages, etc.),

the object can be pre-assigned to the dataset that you want to be created once it is dropped and can be used by the editor. The **Drop Editor** combo box in the relevant table template can be used to select which input component is to have the dropped object pre-assigned to it, and the object will be saved to this input component accordingly.

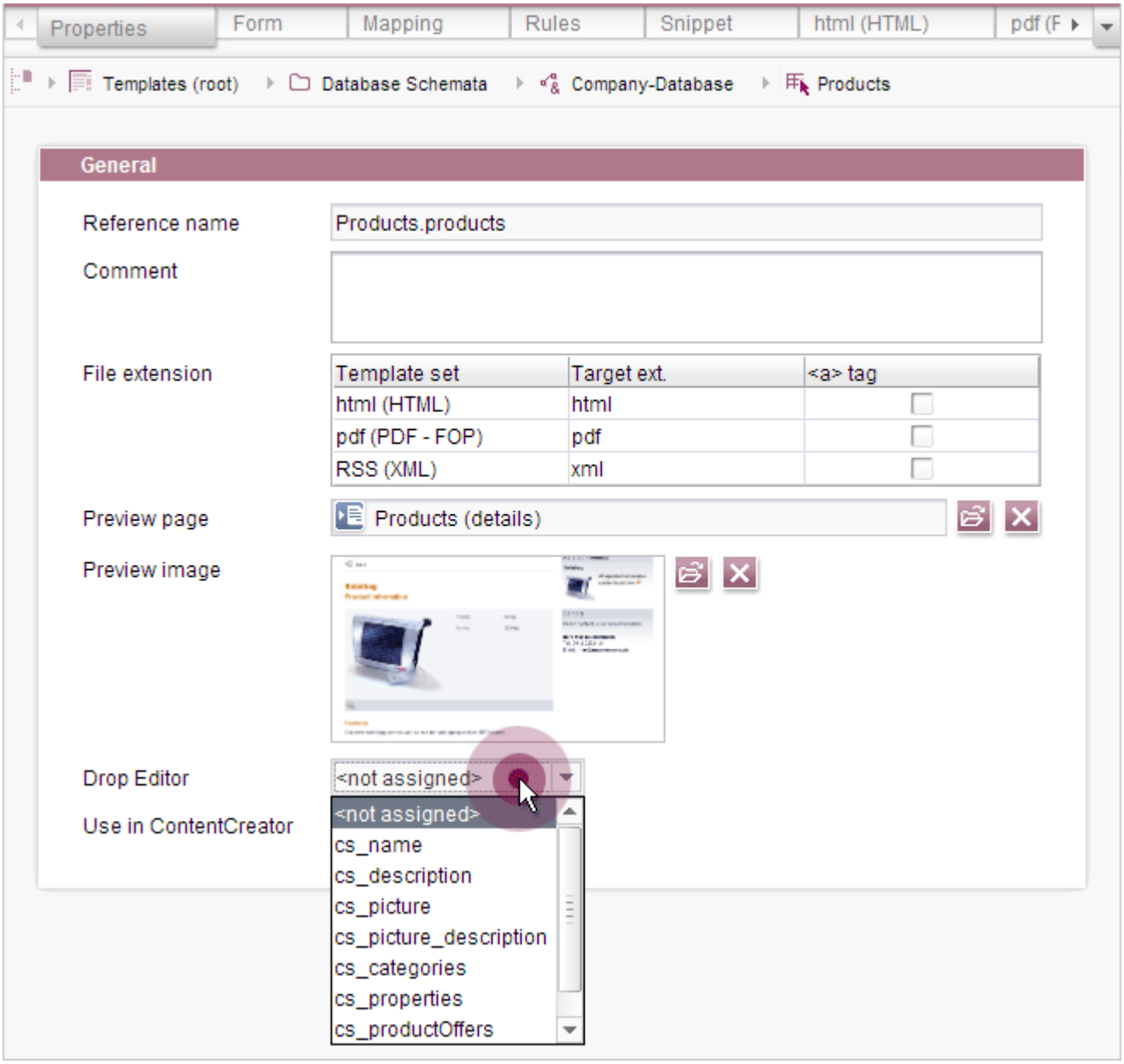


Figure 6-15: Table template – "Properties" tab – Drop Editor

This combo box shows all the identifiers for input components of the following types that have been defined on the "Form" tab:

- FS_BUTTON
- FS_REFERENCE
- FS_LIST
- FS_DATASET
- CMS_INPUT_IMAGEMAP

- CMS_INPUT_DOM
- CMS_INPUT_TEXT
- CMS_INPUT_TEXTAREA

are displayed here.

For example, FS_REFERENCE can be used to incorporate references on pages, page references, and media while FS_DATASET can be used to incorporate datasets. FS_BUTTON can be used in various scenarios, such as when you want to check which type of object may be dropped onto the drop zone (*DROPTYPES* / *MIME*, and *TYPE* tags), e.g., in the case of data originating from customer-specific reports. CMS_INPUT_DOM and other text input components can be used to incorporate dropped text.

In the context of ContentCreator, the table template that is used to create the dataset during a drop operation is the one that serves as the basis for the data source. This data source is specified as follows: via the `content2` parameter within the `fsButton` function that is linked to the FS_BUTTON input component in respect of which the drop was performed. Provided that no rules are violated to prevent saving and that there are no empty mandatory fields, the dataset is created automatically when the object is dropped. During this process, the dropped object is saved in the input component that was selected under "Drop Editor".

If the dataset that is being created contains mandatory fields (e.g., because of rules that are designed to prevent saving or the `allowEmpty="no"` parameter), the form opens after the object is dropped. The dropped object is pre-assigned to the input component selected under "Drop Editor". The editor can choose to accept it or – depending on how the input component is configured – can change/delete it, and enter and save any additional dataset content in the usual manner.

If the dropped object takes the form of a dataset that is compatible with the data source defined via the `content2` parameter, a form opens in ContentCreator with the content of the dropped dataset (copy).

If no corresponding input component has been defined on the Form tab, the selection remains empty (<not assigned>). When other object types are dropped, the table template is not used for the drop operation if <not assigned> has been selected.

For information on the editor view, see also chapter 5.1.4.3, page 38.

6.12.3.4 Links

When FirstSpirit version 5.1 was released it became possible to create links in the rich text editor (CMS_INPUT_DOM) and the rich text editor for tables (CMS_INPUT_DOMTABLE) not only by selecting the  icon, but also – depending on the project configuration – by dragging and dropping elements into the editor. In FirstSpirit version 5.2, input components of type

- CMS_INPUT_IMAGEMAP
- CMS_INPUT_DOM
- CMS_INPUT_TEXT
- CMS_INPUT_TEXTAREA

are now also allowed under the Drop Editor setting. The required input component can be selected from the "Drop Editor" combo box, which is located under "Form variables assignment" on the "Properties" tab for link templates:

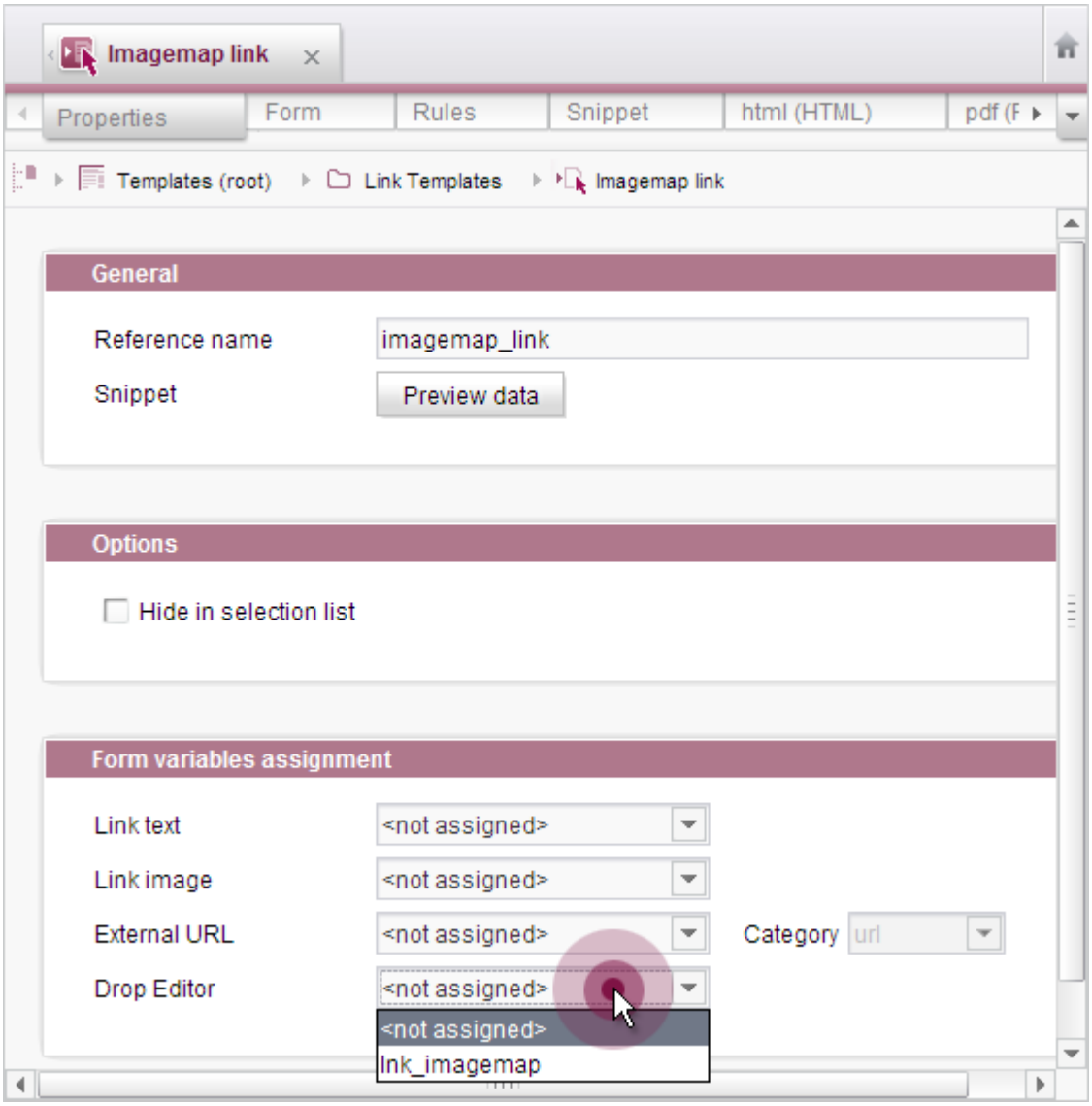


Figure 6-16: Link template – "Properties" tab – Drop Editor

The text input components can, for example, be used to incorporate a link text or link target (URL). This means that the editor can create a link in CMS_INPUT_DOM by dragging text (e.g., the URL of a web page) into CMS_INPUT_DOM.

In the case of image maps, please note that it is not possible to use images from the workstation for the purpose of creating a link. The `DROPTYPES` tag of the FS_BUTTON input component can be used to define or restrict the object types that can be dragged onto the input component with the mouse pointer.

For information on the editor view, see also chapter 5.1.4.4, page 39.

6.12.4 Image cropping function

The image cropping function in ContentCreator is enabled by specifying the `resolution` parameter within the `editorId`-function. The software expects the value to take the form of a resolution reference name (for more information, see ServerManager / Project / Properties / Resolutions / "Name" column). As of FirstSpirit version 5.2, it is now possible to specify more than one resolution for editing by the editor. For more information, see also chapter 5.1.6, page 45.

To enable this, the `resolution` parameter within the `editorId` function now accepts a list of resolutions. The reference names of the required resolutions are specified in square brackets and separated by commas:

```
editorId(..., resolution: ["REFERENCENAME1", "REFERENCENAME2"])
```

The order in which the resolutions are specified is reflected in the image cropping dialog.

In the example below, the image stored in the "st_picture" variable is output using the "TextBildTeaser" resolution. However, the editor is able to edit the "TextBildTeaser" and "Teaser" resolutions in ContentCreator:

```
<img src="$CMS_REF(st_picture, resolution:"TextBildTeaser")$"
  $alt=""
  $CMS_VALUE(editorId(editorName:"st_picture",
                      resolution:["TextBildTeaser", "Teaser"]))$
/>
```

For information on how to define resolutions for a project, see the "FirstSpirit Manual for Administrators", chapter "Resolutions".

For information on how to use the image cropping function in ContentCreator, see the "FirstSpirit Online Documentation", "Template development / Content Highlighting and EasyEdit / Use in projects", section "ContentCreator: Image cropping".

6.12.5 Visualizing report entries in the preview

In ContentCreator, it is now possible to visualize report entries in the preview and filter them according to structural aspects of the project.

To enable this, the `de.espirit.firstspirit.webedit.DataAssociationHandler` class (FirstSpirit Developer API) has been implemented as a new feature. This class can be used to create an association between report objects and objects in the preview. Icons can be shown in the preview to indicate the number of associated report entries.

For example, to display the number of search report entries (i.e., search results/page references) associated with menu items (SiteStoreFolder), you would use the `SiteStoreAssociation` DataAssociationHandler. In this case, the objects would be marked up in the preview using the `editorId()` function.

If you want to apply your own markups, you can use the `dataAssociation()` function:

```
<div$CMS_VALUE(dataAssociation({
            "namespace": "identifier"
}))$></div>
```

To enable access to `DataAssociationHandler`, an interface called `de.espirit.firstspirit.webedit.DataAssociationAgent` is available. This provides methods for determining object assignments and the associated IDs.

A `DataAssociationHandler` is linked to a report via the `DataAccessPlugin` aspect `de.espirit.firstspirit.client.plugin.dataaccess.aspects.DataAssociating`. In addition, a filter is applied here in accordance with a user-selected association.

For information on the editor view, see chapter 5.1.2.2, page 29.

For more information on the `editorId()` function, see the *FirstSpirit Online Documentation*,

- "Template development / Template syntax / Functions / in instructions / editorId"
- "Template development / Content Highlighting and EasyEdit / Use in projects"

for the `dataAssociation()` function

- "Template development / Template syntax / Functions / in instructions / dataAssociation"

6.13 API changes

The FirstSpirit API documentation describes the FirstSpirit interfaces which are used in the templates and scripts to access a huge variety of values, functions, etc. This chapter highlights relevant changes to the API.

6.13.1 Deprecations

The **FirstSpirit Access API** is stable in a minor version series, i.e., the methods available in version 5.2 may change with the next change in major version (to 6.0). This is announced in the corresponding predecessor major version series.

The **FirstSpirit Developer API** is stable in a minor version series, i.e., the methods available in version 5.1 may change with the next change in minor version (to 5.2). This is announced in the corresponding predecessor minor version.

Some methods, classes, and interfaces which had been set to "deprecated" in previous FirstSpirit versions have been removed in FirstSpirit version 5.2, specifically

- FirstSpirit Access API: since ≤ FirstSpirit version 4.2R4
- FirstSpirit Developer API: since ≤ FirstSpirit version 5.1

Also, in version 5.2 itself, methods, classes, and interfaces have been (and are being) set to this state. The methods, classes, and interfaces concerned and the methods, classes, and interfaces that should replace them in each case can be found in the FirstSpirit API documentation. Methods set to the "deprecated" state can still be used, but as they will be omitted in subsequent versions, this is not advisable.

If methods, classes, and interfaces that are no longer supported or are set to deprecated are used in scripts and/or modules, entries are written to log files accordingly (e.g., server log, generation log, etc.). The corresponding log entries should be checked and the cause(s) remedied.

For more information about methods, classes, and interfaces that have been removed from, changed in, or added to FirstSpirit APIs, see also

- FirstSpirit API Documentations
- *FirstSpirit Community*, "Developers / Blog"[5]

6.13.2 Enhancements to the FirstSpirit Access API

The following enhancements, among others, have been made in respect of the Access API:

- **API access to snippet definitions**
  The `Snippet` and `SnippetContainer` interfaces (both `de.espirit.firstspirit.access.store.templatestore` package) can now access snipped definitions ("Snippet" tab in templates). Methods that can be used to change definitions and create new definitions are also available.
- **Identifying the category of external references**
  External links can be categorized via the "Category" combo box on the "Properties" tab for link templates. The `getCategory()` method in the `ReferenceEntry` interface (`de.espirit.firstspirit.access` package) can now be used to identify the category of an external reference.
- **Simplified setting of values (Checkbox, Combobox, Radiobutton)**
  Using the method `set(Object value)` of the interface `FormField` (package `de.espirit.firstspirit.forms`) values can now be set easier into input components of the type
  - CMS_INPUT_CHECKBOX
  - CMS_INPUT_COMBOBOX and
  - CMS_INPUT_RADIOBUTTON
- **API enhancements affecting CorporateContent**
  In the `de.espirit.firstspirit.access.packagepool` package, the
  - `IdTranslator` interface

  has been supplemented with the addition of the
  - `translateIdToSubscriber(Long publisherId)`
  - `translateIdToPublisher(Long subscriberId)`

    methods

  and in the `Subscription` interface the
  - `getElementIdTranslator` (for content packages) and
  - `getTemplateIdTranslator` (for template packages)

[5] https://community.e-spirit.com/community/developer/blog/2015/06/26/api-diff-overview-for-firstspirit-52

methods have been added.
This means that objects in the source project (or "master project) and objects in the target project can now be uniquely referenced: For example, the `translateIdToSubscriber(Long publisherId)` method can identify the ID of the corresponding object in the source project for an object in the target project.

6.13.3 Enhancements to the FirstSpirit Developer API

The following enhancements and changes, among others, have been made in respect of the FirstSpirit Developer API:

6.13.3.1 Multi Perspective Preview

With the release of FirstSpirit version 5.2, the Multi Perspective Preview concept has been carried across to SiteArchitect as well. As part of this process, some new interfaces have been created while others have been deprecated. For detailed information, see chapter 6.7, page 85.

6.13.3.2 Translation help function in ContentCreator

A translation help function has been implemented for FirstSpirit ContentCreator 5.2. This provides a side-by-side view so that content from a form can be easily transferred into another project language. For more information, see also chapter 5.1.3, page 31.

Different projects and different editors may have different requirements as far as the translation process is concerned (for example, which project language is to be the source language, which is to be the target language, which editors are permitted to use the translation help, etc.). For this reason, FirstSpirit offers new interfaces and integration options for ContentCreator so that the translation help function can be flexibly adapted in line with the relevant project requirements. Minimal effort is required to implement the translation help function. It can be integrated into the project precisely where it is needed, e.g., via an InlineEdit button (for more information, see the *FirstSpirit Online Documentation*, "Plug-in development/ContentCreator enhancements/Interactive features/InlineEdit buttons/Function overview"), an FS_BUTTON, or with a report.

As of FirstSpirit 5.2, the InlineEdit button can be used to call a configurable operation. FirstSpirit Developer API entry points:

- `TranslationOperation` (`de.espirit.firstspirit.webedit.server` package):

This interface provides methods for maximum customization of the translation help function in ContentCreator so that it is suitable for the translation processes of the project concerned. This includes a facility for passing a specific source and target language and a feature for integrating a plug-in with external functions to support the desired translation processes within the project. However, for a straightforward side-by-side view that is capable of receiving the editorial content, there is no need to integrate the plug-in. This plug-in can also be used to configure whether content should be translated automatically once at the start.

- `TranslationPlugin` (`de.espirit.firstspirit.webedit.plugin.translation` package):

This interface provides methods for linking external functions within the ContentCreator translation help function, e.g., via a FirstSpirit module.

If a translation module such as this is passed to `TranslationOperation` (as a parameter), an additional button appears when the side-by-side view is called. This button can then be used to start an (optionally automatic) translation process via the external service that has been linked or via an externally installed module e.g., Google Translate service, Trados translation memory). The FirstSpirit ContentCreator 5 plug-in example (downloadable in the *FirstSpirit Online Documentation*, under "Plug-in Development / Examples") offers an implementation of an InlineEdit button which configures and starts the `TranslationOperation` exemplarily.

Translation help without a link to a translation service:

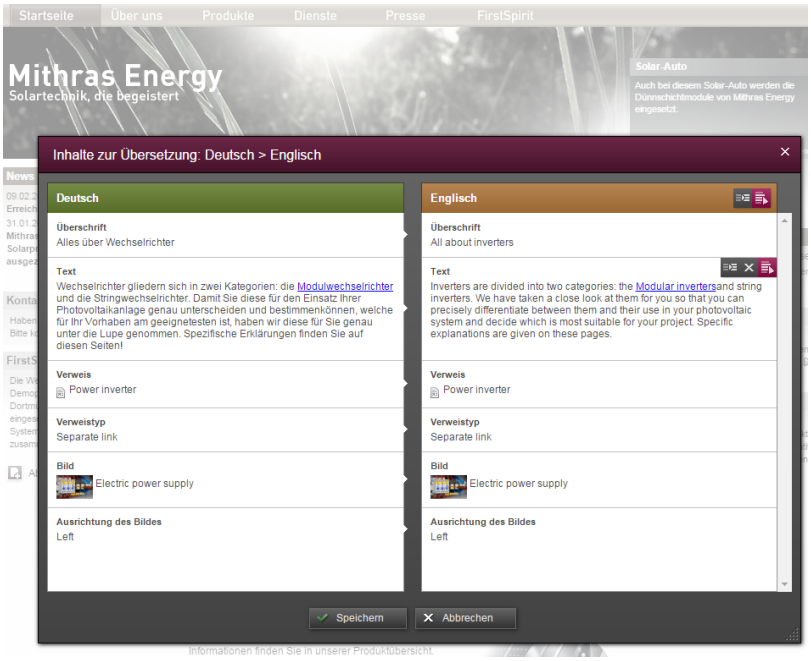


Figure 6-17: Translation help in ContentCreator

The icons can be used to trigger actions involving the transfer and editing of editorial content from one language to another within the side-by-side view.

For more information, see also the FirstSpirit Online Documentation, "Plug-In Development / ContentCreator Extensions / Interactive Features / Translation Help".

6.13.3.3 Assigning file names independently of the URL during generation

FirstSpirit supports many options for generating URLs based on the data entered in the clients by editors. The corresponding FirstSpirit objects can be made accessible at these URLs following deployment. In FirstSpirit version 5.0 specifically, the API was updated and restructured extensively in the area of path generation. The enhancements added and changes made as part of this process also affect the `de.espirit.firstspirit.access.UrlCreator` interface (FirstSpirit Access API), which was replaced by the new `de.espirit.firstspirit.generate.UrlCreator` interface (FirstSpirit Developer API) in FirstSpirit version 5.0.

The `de.espirit.firstspirit.access.UrlCreator` interface deprecated since FirstSpirit version 5.0 was used with the `getFilename` method to generate file names that are different from the URL.

Its replacement, the `FilenameFactory` interface (`de.espirit.firstspirit.generate` package) was introduced with FirstSpirit version 5.2. It supports various methods for generating the required file names.

Example:

- Path in the file system:

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

- URL for the corresponding page:

```
[placeholder]/en/press/pressreleases/overview.html
```

Any existing implementations of the deprecated `de.espirit.firstspirit.access.UrlCreator` interface should be converted to `de.espirit.firstspirit.generate.UrlCreator` as soon as possible.

Example fragment for the module.xml:

```
<module>
          ....
   <components>
      <public>
         <name>Advanced URLs</name>
         <class>de.espirit.firstspirit.generate.UrlCreatorSpecificationImpl
                              </class>
         <configuration>
            <UrlFactory>class name url factory</UrlFactory>
            <FilenameFactory>class name file name factory</FilenameFactory>
                     ....
         </configuration>
      </public>
          ...
```

For more information about using `de.espirit.firstspirit.generate.FilenameFactory` and `de.espirit.firstspirit.generate.UrlCreator`, see also the FirstSpirit Online Documentation, "Advances topics / Generation / Advanced URLs".

6.13.3.4 Implementation of custom schedules (`ScheduleTaskApplication` interface)

FirstSpirit schedule entry planning enables time-controlled execution of schedules (i.e., tasks) at server/project level. Associated actions can be combined in one schedule.

Several system actions are available for standard tasks in the CMS environment, e.g., actions for sending e-mails, running scripts, generating and deploying a project, etc. For easier implementation of custom actions, the new packages listed below have now been created:

- `de.espirit.firstspirit.scheduling`
- `de.espirit.firstspirit.scheduling.agency`
- `de.espirit.firstspirit.scheduling.aspects`

This means that schedules can now be made available via modules with a custom GUI, which makes it easier for users to configure the schedules.

The `ScheduleTaskApplication` interface serves as the starting point. This is used to define the name and the description. It is also where you specify the project context in which the action can be performed (`isApplicable` method). It is the `ScheduleTaskExecutor` interface that provides the actual operation for the action. An associated implementation example is available. This illustrates how to use the API and can be used for testing purposes.

The API within this area is still in the process of development and may be gradually enhanced in subsequent versions.

For more information, see also the FirstSpirit Online Documentation, "Plug-In Development / Server Plug-Ins / Schedule Tasks".

6.13.3.5 Other changes (ContentCreator)

- You can use the new `de.espirit.firstspirit.webedit.DataAssociationHandler` class to assign **report entries to elements in the preview** of ContentCreator. For more information, see also chapter 6.12.5, page 112.
- Within the `Preview` interface (`de.espirit.firstspirit.webedit.client.api` package), the `reload(String htmlId)` method has been enhanced. This makes it possible to trigger a **reload** operation, e.g., after a change has been made in relation to a particular element in the ContentCreator preview. It is no longer necessary to reload the entire page (`reload()` method). Within this context, the ID of an HTML element can be passed as follows:

```
WE_API.Preview.reload("container")
```

Here, "container" is the identifier for an element defined via

```
<div id="container">
```

For more information, see also the FirstSpirit Online Documentation, "Template Development / ContentCreator / Functional Scope".

- Within the `de.espirit.firstspirit.webedit.plugin` package, the `ClientResourcePlugin` interface has been enhanced and within the `de.espirit.firstspirit.webedit.server` package, the `ClientResourceOperation` interface has been enhanced. You can use these to load **resources** (e.g., JavaScript or stylesheet URLs) in ContentCreator. Note: If multiple JavaScript or stylesheet files are specified, their execution is dependent on the browser and so you cannot control the order in which they are executed. See also the *FirstSpirit Online Documentation*, "Plug-In Development / ContentCreator Extensions / Management Extensions / Loading Client Resources".

6.13.3.6 Other changes (SiteArchitect)

- The `de.espirit.firstspirit.ui.operations.ShowReportOperation` interface is new. As a result, reports can now also be opened in SiteArchitect via the API and relevant **parameters passed** as part of this process (similar to the `WE_API.Report.show` method in the `Report` interface (`de.espirit.firstspirit.webedit.client.api` package) for ContentCreator).

6.13.3.7 Other changes (ServerManager)

- Within the `ModuleAdminAgent` interface (`de.espirit.firstspirit.agency` package), the following methods have been enhanced:
  - `isAutostart` (checks whether a module service starts automatically)
  - `isRunning` (checks whether a module service is already running)
  - `setAutostart` (determines whether or not a module service starts automatically)
  - `deployWebApp` (installs ("deploys") web component on the web server)
  - `undeployWebApp` (uninstalls ("undeploys") web component from the web server)
  - `getWebAppConfig` (accesses the configuration directory of a web component)
  - `isTrusted` (checks whether a module can be trusted)
  - `setTrusted` (identifies a module as trusted/not trusted)

In FirstSpirit version 5.2, the `getGlobalWebApps` method now just returns the global web application generated by the user (via FirstSpirit ServerManager / Server properties / Web applications / "Configure global web apps" or `createGlobalWebApp()`). It no longer returns the global system web applications (e.g., fs5root).

For more information on these functions, see also the "Modules", "Trusted modules", "Web applications" and "Web components" chapters of the *FirstSpirit Manual for Administrators*.

6.13.3.8 Other cross-client changes

- A number of new utility classes have been introduced in FirstSpirit version 5.2. These are contained within package `de.espirit.common.tools`:
  - `Images`
  - `Objects`
  - `Streams`
  - `Strings`
- The following packages are new:
  - `de.espirit.firstspirit.client.plugin.dataaccess`,
  - `de.espirit.firstspirit.client.plugin.dataaccess.aspects`
  - `de.espirit.firstspirit.client.plugin.dataaccess.aspects.transfer`

  These can be used to integrate **data object sources** into FirstSpirit and display them there. Sources can either take the form of internal structures (such as data sources/datasets) or external services/applications. These new packages build on the "report plug-ins" that were used in previous versions for the purpose of integrating and displaying external data object sources (and which were implemented by means of the `ReportPlugin` interface in the `de.espirit.firstspirit.client.plugin` package). Any existing instances of the `ReportPlugin` can and should be switched over to the `DataAccessPlugin`. See also the FirstSpirit Online Documentation, "Plug-In Development / Universal Extensions / Data Access" and "Plug-In Development / Universal Extensions / Reports (Legacy) / Functional Overview".
  The use of a DataAccess plug-in that implements the "Reporting" aspect in the client requires an AppCenter license. See also the FirstSpirit Online Documentation, "Plug-In Development / Universal Extensions / Data Access / Use for Reporting" and the FirstSpirit Manual for Administrators.
  The `StaticItemsProviding` interface (in the `de.espirit.firstspirit.client.plugin.dataaccess.aspects`) package means that customer-specific reports can now also be provided with **global actions**

that affect the entire report rather than just one entry. For example, a function for creating new documents can be included in a report that incorporates Google Drive.

- The following methods have been added to the `Common` interface (`de.espirit.firstspirit.webedit.client.api` package):
  - `jumpTo(JavaScriptObject fsid, java.lang.String language)`
  - `getDisplayLanguage()`

  methods have been added.
- The `PropertyProcessing` interface in the `de.espirit.firstspirit.ui.gadgets.aspects` package is new. This allows custom properties to be made available for **customer-specific gadgets** on the basis of rules. For more information, see the *FirstSpirit Online Documentation*, "Plug-In Development / Universal Extensions / Input Components / SiteArchitect", "Plug-In Development / Universal Extensions / Input Components / ContentCreator Gadget / JavaScript Controller (Client)".
- The `ReportContext` interface (`de.espirit.firstspirit.client.plugin.report` package) has been enhanced through the addition of the `repaint` method. This can be used to redraw an individual **entry in a report** (one line).
- The `FsResource.FsIcon` class of the `de.espirit.firstspirit.common` package proveds standard icons for the input component **FS_BUTTON**. This has been enhanced by an icon for new sections starting with FirstSpirit version:  . This icon can be accessed by the key `NEW_SECTION`, e. g.

```
<FS_BUTTON name="button"
...
icon="fs:new_section"
...
</FS_BUTTON>
```

For more information please see also *FirstSpirit Online Documentation*, "Template development / Forms / Input components / BUTTON".

7 New/changed functions for administrators

7.1 Communication between the client and the FirstSpirit server

Internal client-server communication has been optimized as part of FirstSpirit 5.2. When FirstSpirit SiteArchitect is started, the software no longer opens multiple sockets in parallel to enable communication with the FirstSpirit server. Instead, only one individual socket is opened. This change simplifies and speeds up internal processes because it means that multiple sockets no longer have to be opened for each client and authenticated individually.

7.2 Enhanced security measures

In FirstSpirit Version 5.2 and higher, internal communication between the FirstSpirit server components is subject to higher security requirements. This means that all connections to the FirstSpirit server must undergo authentication. In other words, all web applications (and cluster nodes) that communicate with the FirstSpirit server must be authenticated first.

To enable **authentication of the web applications** on the FirstSpirit server, app passwords (application-specific passwords) can be stored in the server properties. These passwords can be configured for all FirstSpirit web applications (fs5root, fs5webmon, etc.) and all cluster nodes. As part of this process, the preconfigured default password must be overwritten. New passwords are generated once only; after this, they are not displayed in plain text again. Therefore, they must be transferred directly to the relevant web application or cluster node as soon as they are displayed in the configuration dialog (see chapter 7.3, page 124).

To ensure that the FirstSpirit start page **remains hidden from search engines**, which makes FirstSpirit servers harder to locate and protects them against attacks, the `robots.txt` file has been introduced as a standard feature of FirstSpirit version 5.2. This is stored in `.../web/fs5root/` and contains the following:

```
User-agent: *
Disallow: /
```

7.3 Authentication for web applications and cluster nodes

With FirstSpirit Version 5.2 and higher, all web applications and cluster nodes that communicate with the FirstSpirit server must undergo authentication. For more information, see also the *FirstSpirit Manual for Administrators*, chapter "Authentication of all internal connections to the FirstSpirit server". For this purpose, a new menu item called "App Passwords" has been made available in ServerManager under "Server properties" so that passwords can be generated and managed for the web applications used:

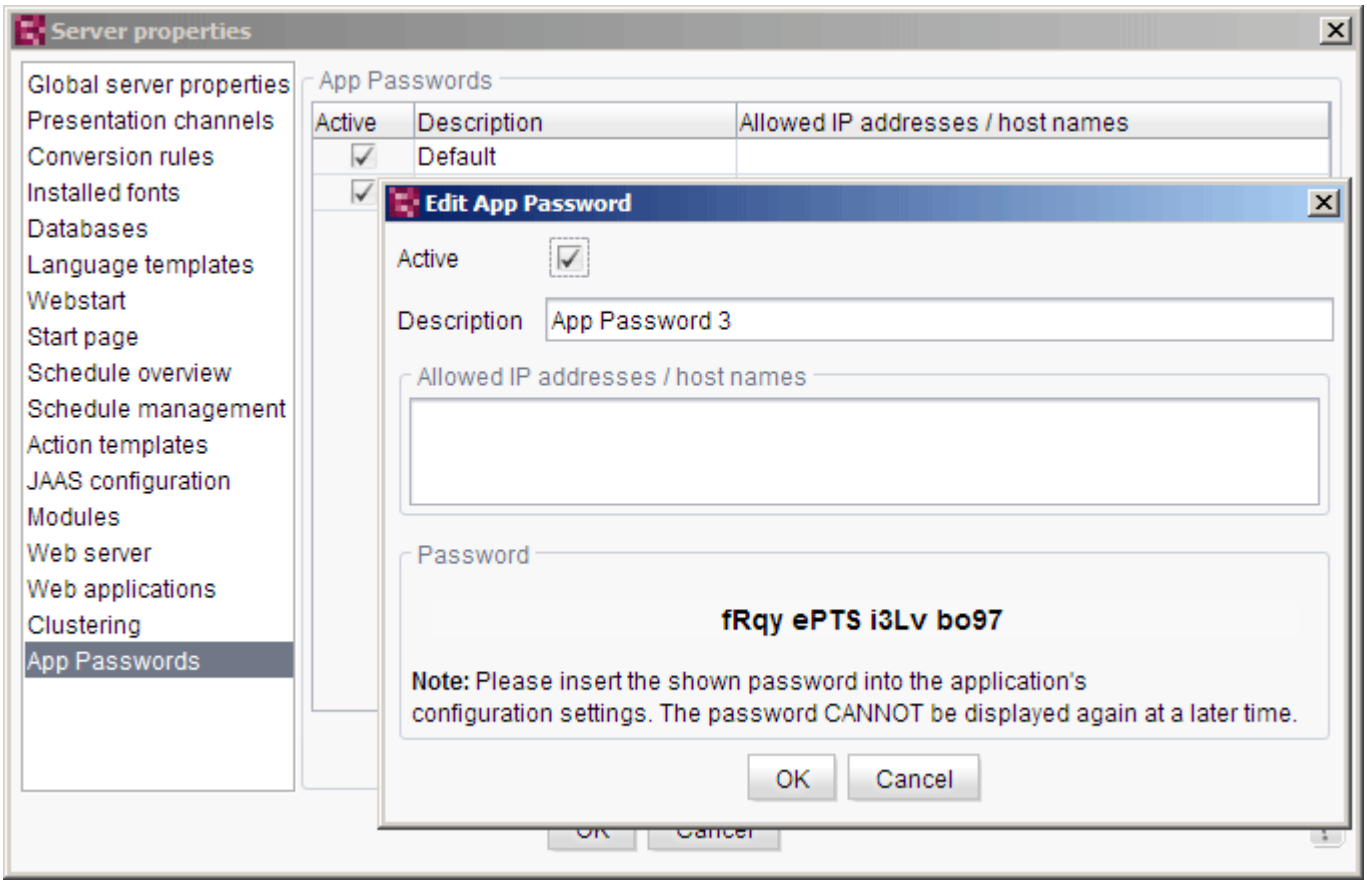


Figure 7-1: Server properties – App Passwords

Initially, only the "Default" password is enabled when you open the "App Passwords" area. The preconfigured default password is kept in the `fs-server.jar` file in encrypted format. This ensures that older FirstSpirit installations remain compatible once they have been updated to FirstSpirit Version 5.2 because it eliminates the need to directly configure all existing web applications and cluster nodes with the new app

passwords for authentication purposes.

To ensure a fully secured connection, the default password for all connections should be replaced with a new app password. Once a new app password has been successfully configured, the default password can be disabled or use of the password can be restricted to particular IP addresses.

It is not possible to delete the default password.

The default password should only be disabled once you are sure that a connection can be successfully authenticated using a new app password. Otherwise, access to the FirstSpirit start page or ServerManager may be accidentally blocked (for more information, see the FirstSpirit Manual for Administrators, chapter "Fixing a faulty configuration: Authentication no longer possible").

The default password is reassigned in conjunction with each FirstSpirit build. Whenever a FirstSpirit update is performed (or fs-server.jars is updated), all web applications have to be updated. This also applies, in particular, to any web applications installed on an external web server (see FirstSpirit Manual for Administrators, chapter "Updating a web component"). This will ensure that the default password used in the `fs-server.jar` *(FirstSpirit server) and* `fs-webrt.jar` *(application server) files is identical. Otherwise, authentication will no longer be possible after the update is performed.*

New app passwords are generated by pressing the "Add" button. They are generated on a one-time basis and are not saved. Therefore, the password should be transferred directly to the configuration for the web applications and cluster nodes (any white spaces in the password are merely intended to improve legibility and can be removed if necessary). The generated password cannot be displayed again subsequently.

The app passwords can only be used for direct socket connections (not for http connections). In addition, the app passwords cannot be used for conventional user authentication processes.

For detailed information on generating and using app passwords, see also the *FirstSpirit Manual for Administrators*, chapter "App passwords".

7.4 The FirstSpirit Launcher

Up until now, the only way to start and update (roll out) FirstSpirit SiteArchitect (and ServerManager) was to perform the process in a browser via the FirstSpirit start page using Java Web Start technology. This meant that a Java Runtime Environment (JRE) had to be present on the FirstSpirit editor's workstation[6]. Consequently, Java had to be installed on the system *and* Java Web Start had to be enabled in the browser.

Security gaps have repeatedly come to light in many Java versions and, for this reason, the software is no longer to be installed on all workstations without restriction. Alongside the problems affecting security, the administration and maintenance overheads for Java must not be underestimated. Java must be rolled out on all workstations and regular updates are required. Furthermore, incompatibilities and version conflicts can occur if the workstation concerned is running other software components that rely on different versions of Java.

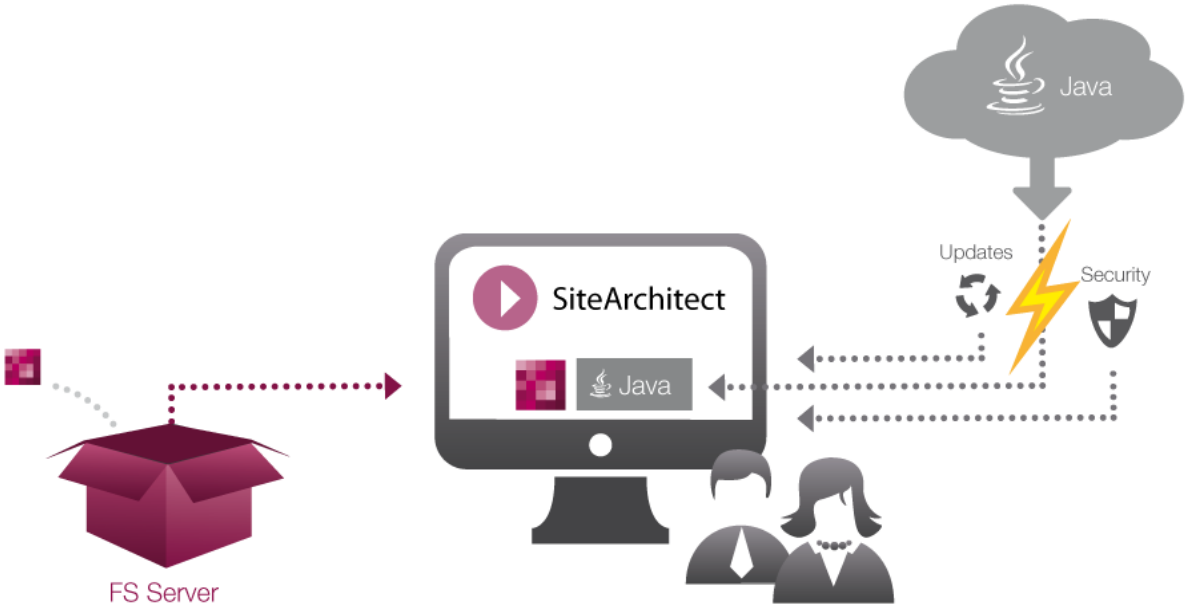


Figure 7-2: Starting SiteArchitect via Java Web Start

[6] For more information, see the FirstSpirit Technical Data Sheet.

The purpose of the FirstSpirit Launcher is to offer an alternative to the technology that has been used so far and to bypass the constant problems that are encountered in connection with a Java-based infrastructure. The FirstSpirit Launcher has not actually eliminated the need for a Java Runtime Environment (JRE). However, when the Launcher is used, SiteArchitect and ServerManager are still able to run even if Java cannot/should not be made available on the workstations of the FirstSpirit users. For the process to work, the FirstSpirit Launcher rolls out its own JRE during installation. As a result, FirstSpirit is no longer dependent on having a Java version pre-installed on the local workstation of the FirstSpirit user.

Advantages of using the FirstSpirit Launcher:

+ Security: Java is no longer required on the browser page. This eliminates a potential security vulnerability in terms of outside attacks.

+ Update: The amount of administration and maintenance work can be significantly reduced because no JRE is installed on the workstations and so they no longer have to undergo regular Java updates. The Java version of the Launcher is now updated automatically as part of the FirstSpirit server update process.

+ Compatibility: The Java version used within the Launcher is selected and extensively tested by e-Spirit. As a result, it is possible to prevent the use of Java versions that have already been identified as problematic. Another positive aspect of the Launcher is that it eliminates the possibility of incompatibilities or version conflicts with other Java products that are installed on the system.

From a technical perspective, this solution is highly sophisticated: Given that the browser itself no longer supports Java, all the information provided by FirstSpirit (SSO, login process, project information, etc.) has to travel from the browser level to the operating system level of the local workstation via a different route. By definition, this route is highly secure because it is recognized as the potential gateway for outside attacks.

Solution: A file system extension is registered by means of a Windows native implementation. Then, a `.fslnch` configuration file (text file) is generated from the start page and downloaded from the browser (which is regarded as secure). This configuration file is then linked to the FirstSpirit Launcher and its first task is to ensure that the JRE and the JAR are downloaded from the FirstSpirit server (from the `fs5root` directory). The Launcher uses the configuration file to find out where these files are located. Then, the FirstSpirit applications can be started in the usual manner (SiteArchitect, ServerManager).

Restrictions:

- The FirstSpirit Launcher is only available for Windows systems.
- Currently, the FirstSpirit Launcher does not support all network settings which are possible with Java Web Start (e.g., no proxy setting with PAC[7] configuration).
- For technical reasons, it is not possible to use the "Switch projects" function ("Project" menu) in SiteArchitect when working with the FirstSpirit Launcher.

For more information and limitations, see the "FirstSpirit Launcher" chapter of the *FirstSpirit Installation Instructions*.

7.5 Using start page configuration instead of Java Web Start technology

Another way to start FirstSpirit SiteArchitect and ServerManager without having to use Java Web Start technology is to configure the `fs-server.conf` file accordingly. The following parameters are available for this purpose:

`startpage.webstart.url`: This parameter allows you to define a base URL, which is called via the FirstSpirit start page whenever SiteArchitect or ServerManager are started. The parameters that are required to start the client (the UTL parameters) are appended to this URL. The parameters take account of the connection settings that have been configured and are active in each case (FirstSpirit start page / "Connection settings", FirstSpirit ServerManager / "Server properties" / "Webstart" or FirstSpirit ServerManager / "Server properties" / "Start page"). This means, for instance, that you can use your own Java start system instead of the Java Web Start technology.
Example: `startpage.webstart.url=http://MyServer:8080/MyStartPage.jsp`

`startpage.logout.url`: This parameter allows you to define a URL to which users are directed when they log out of FirstSpirit (FirstSpirit start page / "Logout").
Example: `startpage.logout.url=http://www.e-Spirit.de`

[7] Proxy auto-configuration

If you use the `allowedRedirectHosts` parameter in addition to the `startpage.webstart.url` and `startpage.logout.url` parameters, you must take care to ensure that the relevant URLs are included in the list for `allowedRedirectHosts` as well (or that the `FS_SERVER` or `ALLOW_ALL` values are set). Otherwise, an "HTTP ERROR 500" error will occur on start-up. Logout triggers an actual logout process and the login page appears.

For more information, see the following chapters of the *FirstSpirit Manual for Administrators*:

- "Area: Misc"
- "Area: JumpToServlet and ContentCreator ForwardAction"
- "Webstart"
- "Start page"

7.6 Multi Perspective Preview in SiteArchitect

As Internet-enabled mobile devices such as notebooks, tablet PCs, and smartphones become more and more widespread, website designs need to be more and more flexible, with content which can be displayed perfectly on different display geometries and in different resolutions. That is why FirstSpirit version 5.1 introduced an easy way for editors to check what website content looks like and how well it can be navigated with a variety of display sizes in ContentCreator, while also allowing content, layouts, and images to be perfectly adapted for the output device concerned. Along with size considerations, other aspects can also be taken into account, e.g., previews for specific user groups ("Multi Perspective Preview", "MPP").

With the release of FirstSpirit version 5.2, the Multi Perspective Preview concept has been carried across to SiteArchitect as well. For more information, see also chapter 5.2.1, starting on page 50.

The configuration process is the same for both ContentCreator **and** SiteArchitect, and relies on the following options:

- Configure preview viewports
- Preview parameters

These can be found in FirstSpirit ServerManager under "Project properties / Options":

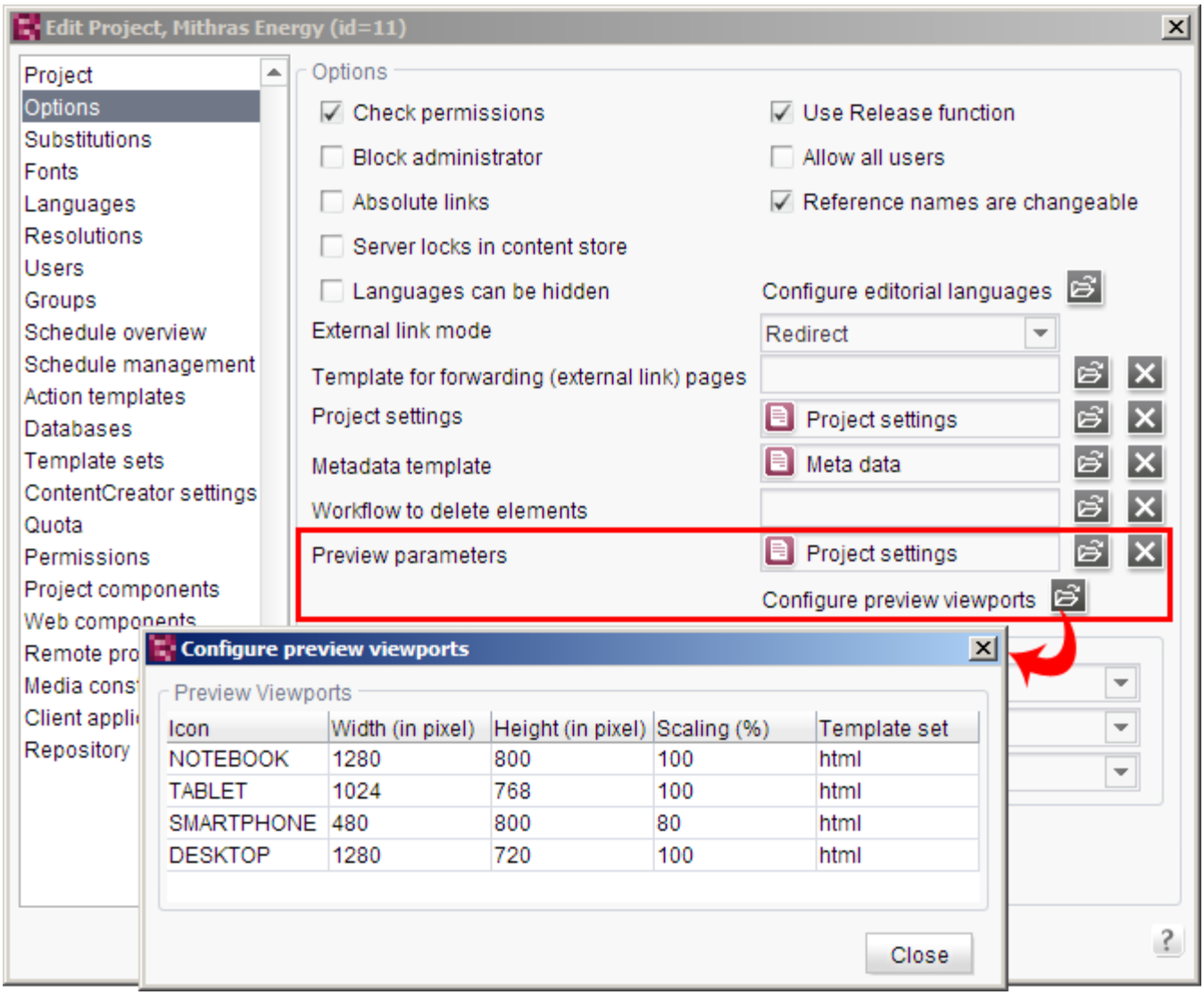


Figure 7-3: MPP configuration

For detailed information on the configuration options, see:

- Chapter 6.7, page 85
- *FirstSpirit Manual for Administrators*, chapter "Options".

7.7 Changes to schedule management

Implementation of custom schedules (ScheduleTaskApplication interface), e.g., for the generation and deployment of a project, for server maintenance, to run scripts, etc.

The schedule management interface in ServerManager ("Server/Project properties / Schedule management / Add/Edit / Actions / Add") has been redesigned as part of the new "Implementation of custom schedules (`ScheduleTaskApplication` interface)" functionality (chapter 6.13.3.4, page 118).

Schedules that have already been completed or are planned are displayed in ServerManager in table format under the "Schedule overview" menu item. Previously, all schedules within the selected time frame were displayed. This could lead to performance problems and dropped connections. With FirstSpirit version 5.2, now only 40 schedules are displayed initially. More entries can be loaded by pressing the "Show more" button (see Figure 7-4). If there are fewer than 40 entries for the selected time frame, the button is grayed out.

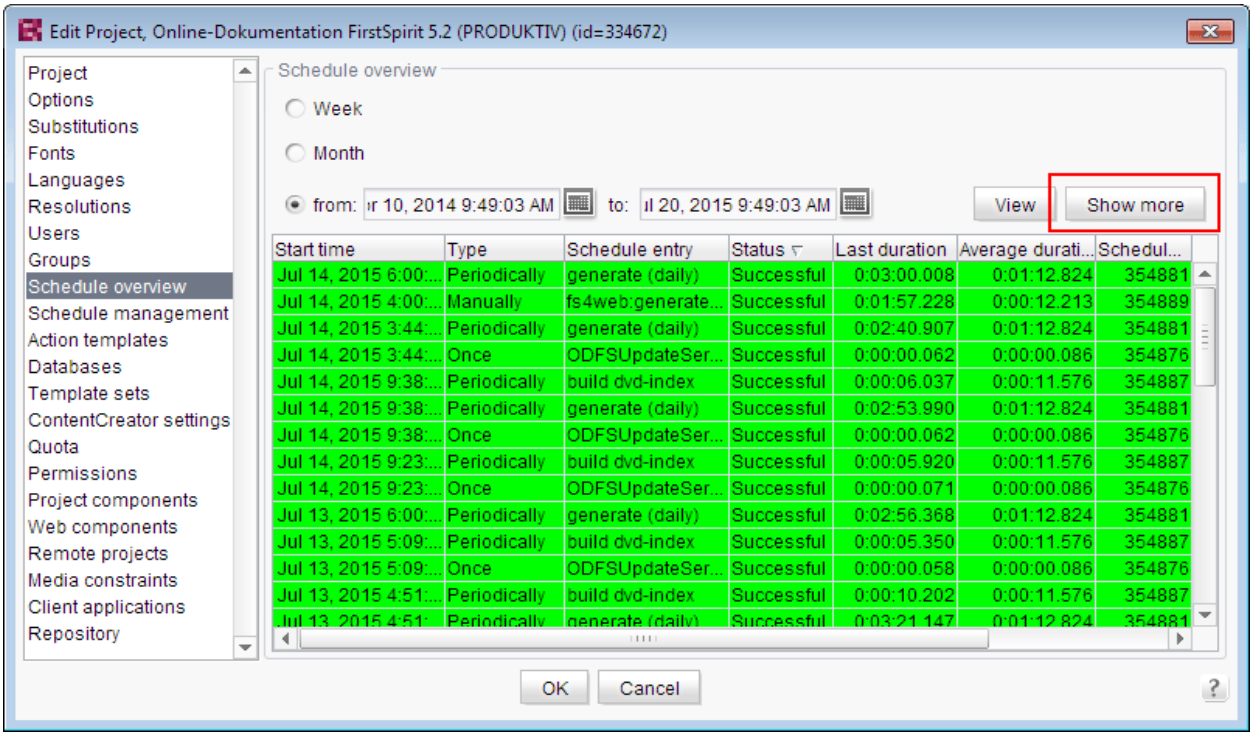


Figure 7-4: FirstSpirit ServerManager – Schedule overview

For more information on FirstSpirit schedule management, see the "Schedule entry

planning" chapter of the *FirstSpirit Manual for Administrators*.

7.8 Changes concerning databases

In FirstSpirit version 5.0, a new ID format was introduced for datasets so that they could be more easily identified. The new ID format is based on the UUID standard[8]. In the case of existing projects, the format is switched over automatically as soon as datasets are edited in the FirstSpirit project. This means that newly added or changed and saved datasets will automatically receive a **GID** ("Global ID") in version 5.0 and higher. As of FirstSpirit 5.2, the GID column within DB tables is now also indexed automatically (DBMS indexing for table columns). This only applies to new or newly imported FirstSpirit projects. In the case of existing projects, the database administrator must take the appropriate steps if GID column indexing is required. GID column indexing can be deactivated with

```
jdbc.CREATE_INDEXES=false
```

in the database configuration (via FirstSpirit ServerManager) if you want to revert to the previous behavior.

See also the "Database connection" chapter of the *FirstSpirit Manual for Administrators*.

The interpretation of the optional database parameter **`jdbc.POOLMAX`** has also changed: FirstSpirit uses a separate pool instance for every application, e.g., a schema node in a project, a deployed web application (e.g., FirstSpirit DynamicDatabaseAccess), etc. Although `jdbc.POOLMAX` previously defined the maximum number of unused DB connections that can remain in the pool, in FirstSpirit version 5.2 and higher, the parameter now limits the maximum number of connections that can be made to the database.

`jdbc.POOLMAX` must be at least the value of `jdbc.POOLMIN` + 4 (the default value of `jdbc.POOLMIN` is 5). If it is not, the value is adjusted automatically and a corresponding warning is logged.

When the value of `jdbc.POOLMAX` is reached, there is a default wait time of 5 seconds for a free connection. After this, an exception occurs at the user (e.g., client,

[8] http://docs.oracle.com/javase/1.5.0/docs/api/java/util/UUID.html

generation). The wait time for a free connection is configured with the `jdbc.FETCH_CONNECTION_TIMEOUT` parameter (the value is set in milliseconds).
For more information, see the "Description of optional parameters" chapter of the *FirstSpirit Manual for Administrators*.

7.9 Extension of error reporting

FirstSpirit 5.0 has created an infrastructure for Internet-based error reporting. Error reports can be sent to eSpirit via a web interface. Access data and an Internet connection are required. Access data can be obtained from Technical Support.

Automatic reporting is deactivated by default. If it is activated, the following information is included in the data that is sent to e-Spirit:

- Software status: FirstSpirit version, installed modules, license
- System environment: operating system, Java version, etc.

Whilst every effort is made not to send project-specific content, this cannot be guaranteed.

*Users are **not** entitled to troubleshooting support for errors that are detected! For troubleshooting support provided in the context of software maintenance, all errors must be reported via the specified method.*

With FirstSpirit version 5.2, information that users can enter themselves (text and screenshot) can be sent along with the information that is collected automatically by the system (e.g., the technical error message, software and hardware environment, etc.). See also chapter 5.2.7, page 59 for more information.

These error reports are stored in the "~/log/report" directory on the FirstSpirit Server. This information can also be sent to the manufacturer e-Spirit (server-based action "Report compaction" in schedules, see the *FirstSpirit Manual for Administrators* for more information).

Text entered by the user (in the Figure 5-24: Adding additional information dialog) is displayed in the "Systemreport.html" file under "Custom Description", for example. The screenshot (in the Figure 5-24 "Capture image") is displayed under "Binary".

7.10 Log rotation of the Java VM garbage collector ("fs-gc.log")

In earlier versions of FirstSpirit, the Java VM garbage collector was logged by default on FirstSpirit server pages in the "fs-gc.log" file (inside directory "~/log/") and in a file called "fs-gc.*.log" (* stands for the date and time of the first file entry). If a fixed file size of 5 MB was reached, the current log file was compressed and archived to a file called "fs-gc.*.log.gz", where * once again stands for the date and time of the first entry.

As of FirstSpirit version 5.2, the names of the log files for newly installed FirstSpirit servers are no longer generated using the date and time; instead, the files are consecutively numbered, starting with 0 ("fs-gc.log.0", "fs-gc.log.1", etc.). With the default configuration, the tenth file overwrites the first file created, i.e., the one called "fs-gc.log.0" ("log rotation"). This rotational configuration prevents the number of log files from constantly increasing and taking up more and more of the disk space.

If the FirstSpirit server is restarted, the file names start again from "fs-gc.log.0", which means that the log files created prior to the restart are overwritten.

Within this context, the following parameters have been incorporated into the "fs-wrapper.conf" and "fs-wrapper.slave.conf" configuration files (inside directory "~/conf/"):

```
wrapper.java.additional.41=-XX:+UseGCLogFileRotation
wrapper.java.additional.42=-XX:GCLogFileSize=10M
wrapper.java.additional.43=-XX:NumberOfGCLogFiles=9
```

The `-XX:GCLogFileSize` parameter can be used to set the maximum file size that can be reached before a new file is created. The minimum value is 8 KB (default value: 10 MB).

The `-XX:NumberOfGCLogFiles` parameter can be used to set the maximum number of log files that should be retained (default value: 9).

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

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

If you want the system to revert to its previous behavior, these three parameters can be commented out in the "fs-wrapper.conf" configuration file.

If you want to use the log rotation feature on existing FirstSpirit servers, the three parameters can be entered in the configuration files manually.

8 New/changed functions in modules

New methods for implementing customer-specific modules have been incorporated into the FirstSpirit Developer API. See chapter 6.13, page 113 for more information.

8.1 New core modules

The following modules are new in FirstSpirit version 5.2:

- "FirstSpirit Agency Support" module (`fs-agencysupport.fsm`): See chapter 6.2, starting on page 69

8.2 FirstSpirit ContentTransport: Transportation of project properties

With the release of FirstSpirit version 5.2, the "FirstSpirit ContentTransport" functionality not only allows you to transport project content that has been entered using SiteArchitect and/or ContentCreator, but also project properties as well. In principle, this process works in the same way as an export/import performed using the "External synchronization" functionality. See also chapter 6.6, starting on page 84.

In the case of ContentTransport, the project properties to be transported can be configured with the `configurePropertiesTransport` method, `FeatureModel` in the `de.espirit.firstspirit.feature` package in the FirstSpirit Developer API.

For more information on the "FirstSpirit ContentTransport" functionality, see also the "FirstSpirit ContentTransport" module documentation.

8.3 FirstSpirit DynamicPersonalization: Secure use of cookies

You can use the "FirstSpirit DynamicPersonalization" module to personalize how FirstSpirit content is displayed. The module offers various options for logging on, performing authentication, and reading out user-specific information. Depending on the configuration, authentication can take place via a cookie that is generated by FirstSpirit.

As of FirstSpirit version 5.2, it is possible to set the "Secure" and "HttpOnly" attributes (also known as "flags") for these cookies to ensure better protection of the data stored on the client side. This feature involves checking the "SSO Cookie Secure" and/or "SSO

Cookie HttpOnly" boxes in the "FIRSTpersonalization" web component:

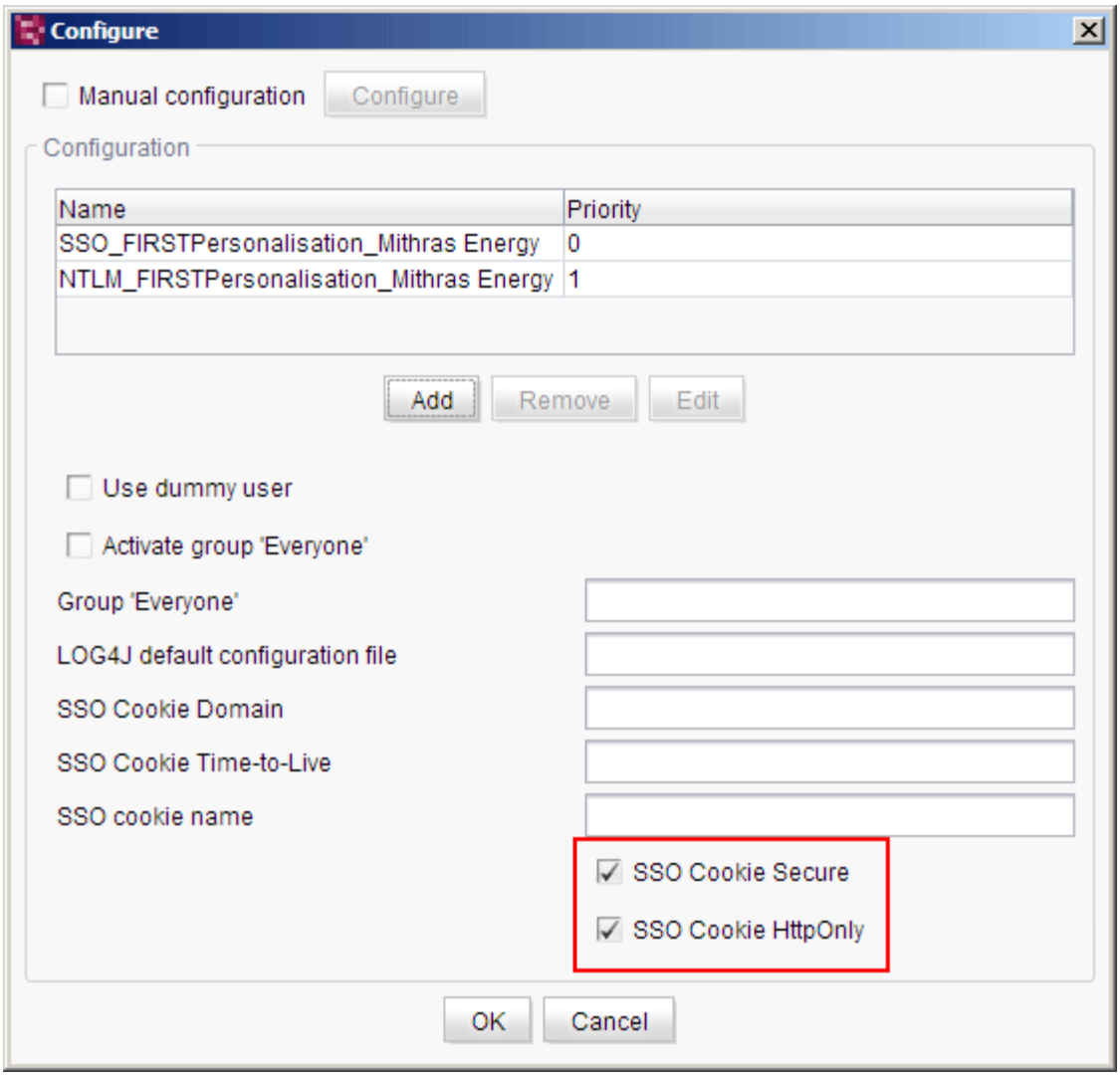


Figure 8-1: Module configuration, "FIRSTpersonalization" web component

SSO Cookie Secure: Checking this box activates a security setting for the cookie. This then means that the cookie can only be sent to the server over HTTPS.

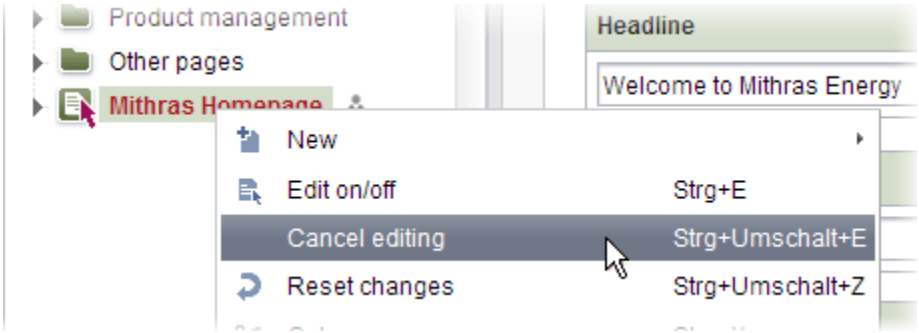
SSO Cookie HttpOnly: Checking this box prevents access to the cookie via JavaScript. This setting can potentially provide protection against XSS provided that the browser used supports the HttpOnly attribute.

For more information, see also the *FirstSpirit DynamicPersonalization* module documentation, chapter "Configuring the web application".

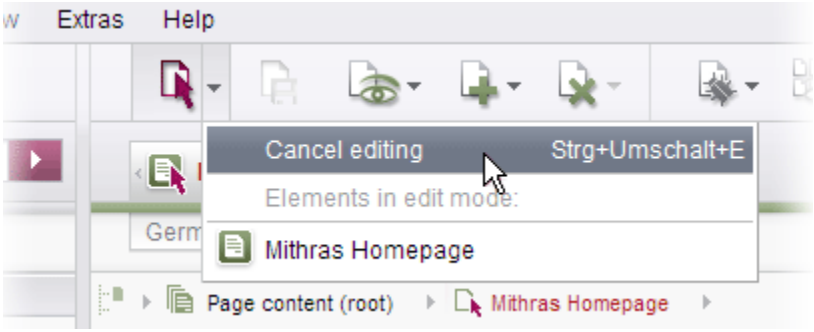
9 Appendix

9.1 Changes in software behavior

- **Google Chrome:** The BETA stage for testing Google Chrome as a browser engine for the integrated preview in FirstSpirit SiteArchitect is now complete. It is being officially released as part of FirstSpirit version 5.2.
- **Cancel editing:** The "Cancel editing" function, which you used to access in SiteArchitect by going to the context menu for objects in the "Extras" area, is now located directly below the "Edit mode on/off" function:

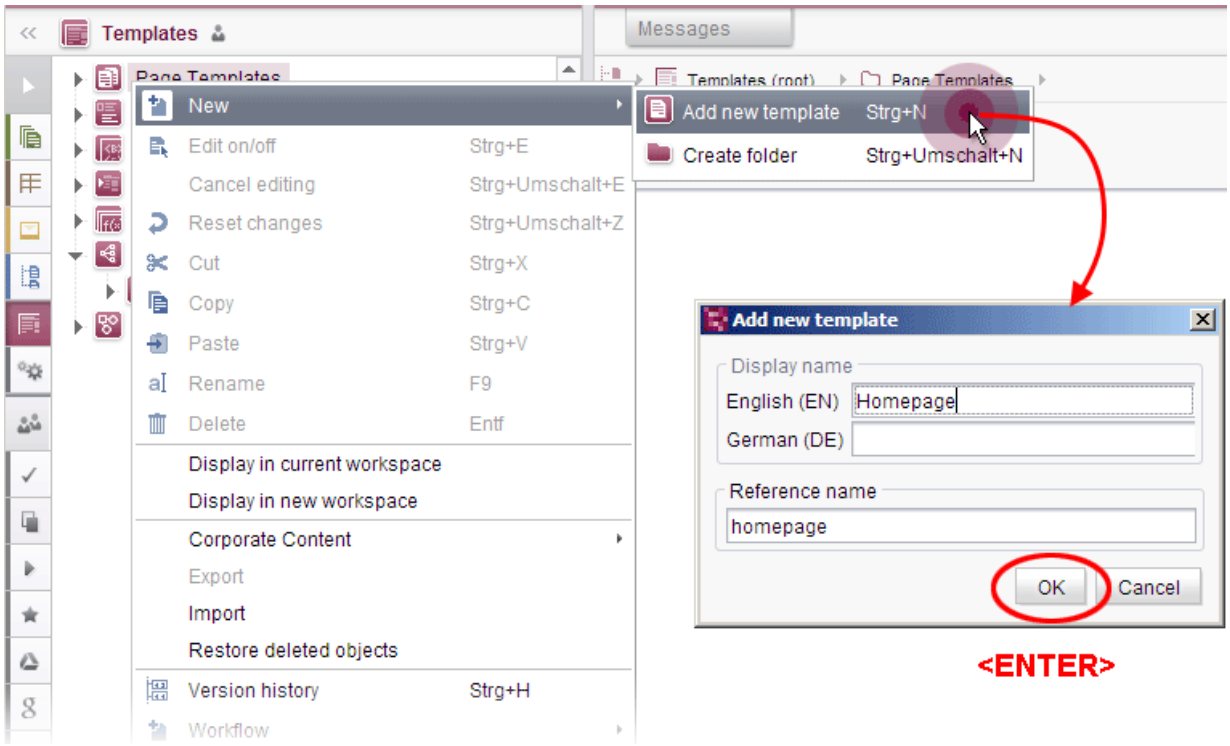


It can also be accessed via the "Switch to View mode" icon in the horizontal tool bar:

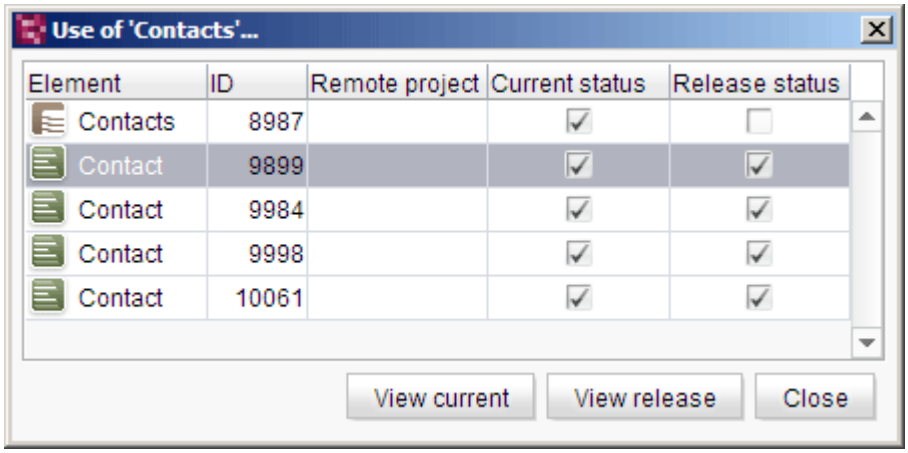


- **Switch project:** The "Switch project" function ("Project" menu) is not active in SiteArchitect when using the FirstSpirit Launcher (see chapter 7.4, page 126).
- **Definition of default values in templates:** It is no longer possible to define default values for templates by using the "Default values" button under "General" on the "Properties" tab. Instead, you can use the "Default values" function (icon) that was introduced as part of FirstSpirit Version 5.1 and which can be found on the "Form" tab.
- **Creating objects:** New objects in SiteArchitect can now be created more quickly using the keyboard: All the dialogs for creating new objects (folders, pages, and sections in the Page Store, media, templates, etc.) – e.g., dialogs accessed via the

"New" context menu – can now be confirmed directly with <ENTER> as soon as the display name has been entered in one language or the reference name has been entered. It is no longer necessary to switch to the <OK> button first via the keyboard or with the mouse. If you need to specify different values in different languages and, where applicable, to serve as a reference name (provided this option has been configured), you can use <TAB> or the mouse to switch to the other fields first.



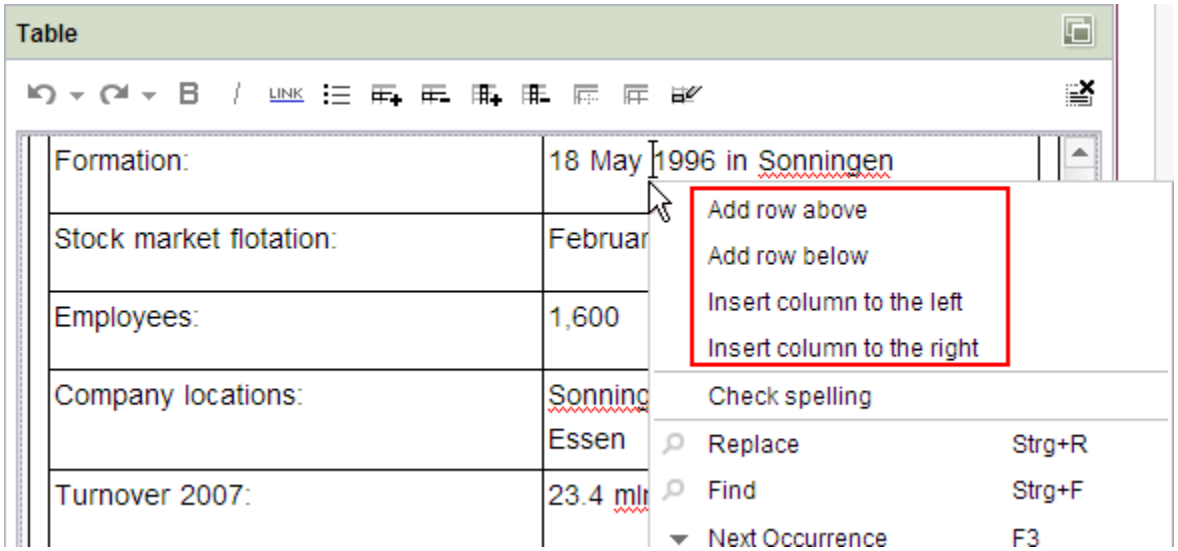
- **Show usages:** Select the "Extras / Show usages" function from the context menu (Ctrl + U) in SiteArchitect to determine where (else) in the project the corresponding object is referenced. A window opens in which the reference points are listed. In the case of templates and datasets, the referenced objects can now also be displayed in both the current and released states, if applicable. For example:



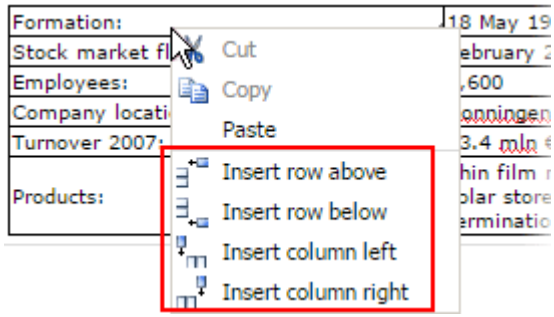
Click the "View current" or "View release" button to display the object in the corresponding state. Alternatively, the current state can be displayed by double-clicking the corresponding reference; if the object is only available in the last released state, it is displayed in this state in response to double-clicking.

The current state of the referenced elements is taken into account for other other object types in this dialog. Click the "View" button to switch to corresponding referenced element.

- **Default value for lists in CMS_INPUT_DOM/_DOMTABLE:** The `listDefaultConfig` parameter can be used to configure default values for lists. With FirstSpirit version 5.2, the default behavior for `style` **in SiteArchitect** has changed: If a value is not set for the `style` attribute, `style="1"` (bullets or image from the Media Store defined with `mediaref` in `listDefaultConfig`) is always used by default. Prior to FirstSpirit version 5.2, the default value was `style="0"` (dashes). You can recreate this behavior in FirstSpirit version 5.2 by setting `listDefaultConfig="style=0"`.
- **Adding new rows/columns to tables:** In **SiteArchitect and ContentCreator**, the "Add row above", "Add row below", "Insert column to the left", and "Insert column to the right" functions have been added to the context menu for tables (CMS_INPUT_DOMTABLE and CMS_INPUT_DOM):



and



- **Selecting datasets in FS_LIST**: Lists of datasets can be created with the "DATABASE" FS_LIST type. If the selection of datasets with

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

is permitted, a dialog window in which the required dataset can be selected opens by default in SiteArchitect (corresponds to `<PARAM name="select-ui">dialog</PARAM>`) when you click the "Add" icon. In FirstSpirit 5.2, a "drop zone" has been added to this dialog in which a list of the selected datasets is displayed and multiple selections can be made.

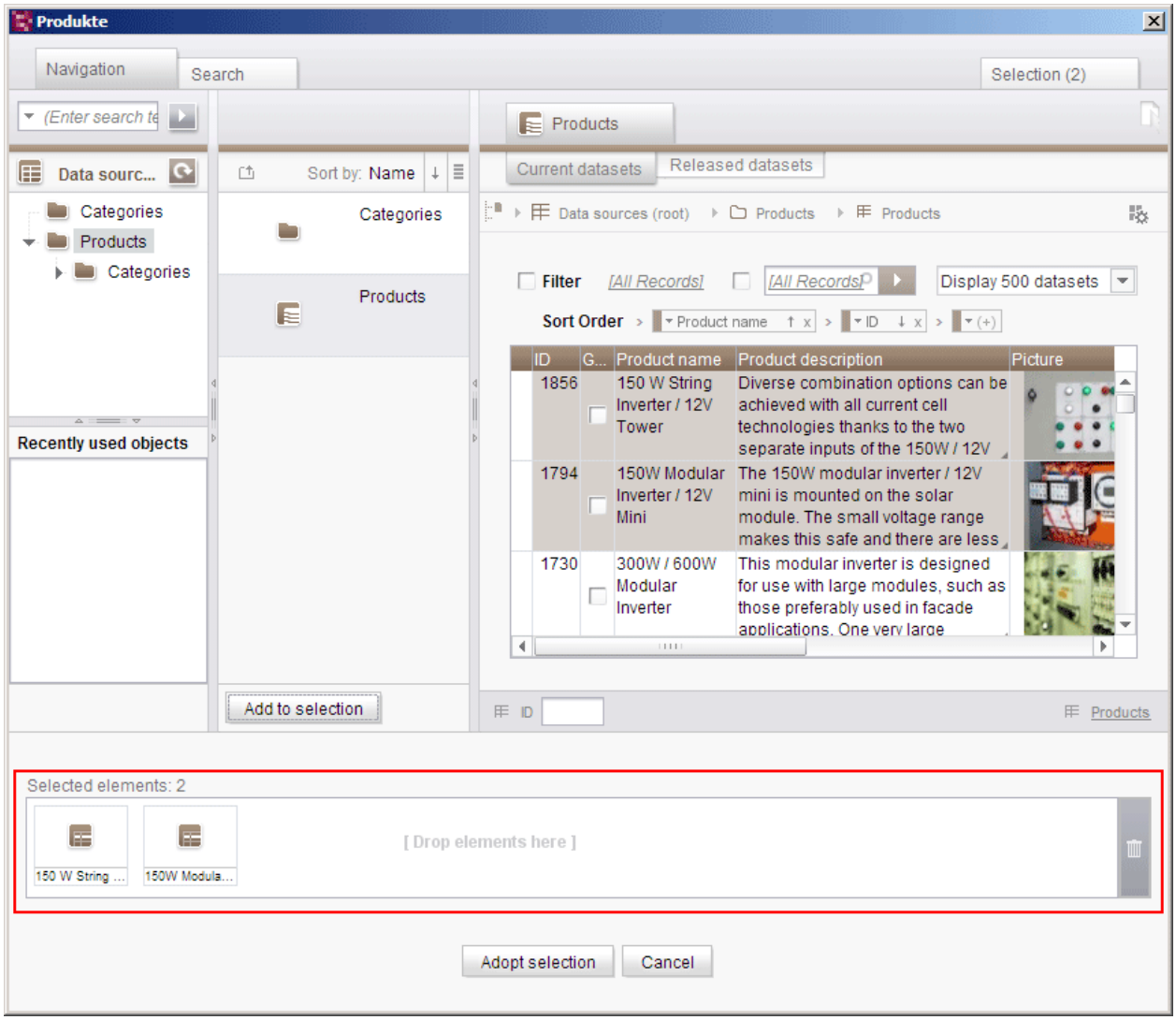
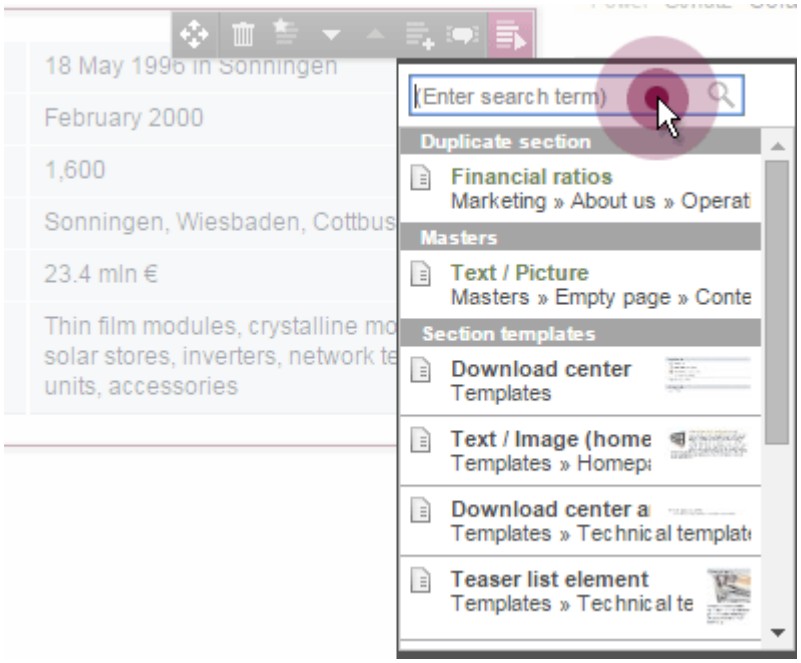


Figure 9-1: FS_LIST, selection of multiple datasets

The datasets can be selected either with drag-and-drop or by clicking the "Add to selection" button under "Selected elements". To select multiple datasets, press and hold down the <CTRL> or <SHIFT> key. The collection of all selected datasets can then be added to the input form by clicking the "Adopt selection" button. This dialog is also used if `<PARAM name="select-ui">list</PARAM>` or `<PARAM name="select-ui">popup</PARAM>` is configured and multiple data sources are available for the table template defined with the TABLE tag (filtered view). For more information, see

- FirstSpirit Online Documentation, "Template development / Forms / Input components / FS_LIST / DATABASE", <ACTION/> tag
- FirstSpirit SiteArchitect manual, "Data lists" and "Selection dialogs" chapters

- **Rules / "query":** As with CMS_INPUT_COMBOBOX, query-based population via rules is now also supported by CMS_INPUT_RADIOBUTTON, CMS_INPUT_CHECKBOX, and CMS_INPUT_LIST. For more information, see also the *FirstSpirit Online Documentation*, page "Template development / Rules / Form properties <PROPERTY/> / Property VALUE", section "Setting dynamic values using a database query" and chapter 6.3.7, page 63.
- **Code completion for CMS_COMMENT:** The code completion feature for the CMS_COMMENT design element that allows you to comment out individual passages in the Form area of SiteArchitect (e.g., input components that are temporarily not required) has been optimized.
- **"Duplicate section" function:** The "Duplicate section" icon has been removed from ContentCreator. To duplicate a section, you now have to go via the "New section" icon instead. A window will then open, allowing you to select the "Duplicate section" function.



- **Icons in ContentCreator:** Content is often edited directly in the ContentCreator preview. Icons with various functions are available to facilitate this process.

The number of icons and the order in which they appear may vary from one project to another. In previous versions, the icon that appeared on the right was always blackberry-colored, but now it is the "Edit" icon that has the blackberry color applied to it by default. If, for example, the "Edit" icon is hidden because of the permission definition, the blackberry color is not applied to any of the other icons instead.

- **Configuring resolutions:** In FirstSpirit, different resolutions can be defined for a project in ServerManager. The system can automatically calculate the resolution for each image uploaded to the project in accordance with the resolutions defined for the project. Alternatively, it is possible to define an image section for each resolution or to upload a custom image. In the ServerManager table view that shows the resolution information (available under "Project properties / Resolutions"), it is no longer the description that is displayed. As of FirstSpirit version 5.2, the comment is displayed instead. For more information, see also the *FirstSpirit Manual for Administrators*, chapter "Resolutions".
- **Standard version Oracle Berkeley DB:** With FirstSpirit version 5.2, version 5.x of the Oracle Berkeley DB is used by default for projects which are created as new or imported on a FirstSpirit Server version 5.2.
- **Starting external processes in POSIX environments:** Starting external processes from FirstSpirit tasks using `de.espirit.firstspirit.agency.ProcessAgent` (see FirstSpirit Developer API) has been optimized such that processes are started using `posix_spawn()` if this function is supported by the operating system. This change allows external processes to be started without increased virtual memory requirements, and special swap space configuration for FirstSpirit is no longer required. This change has no effect on FirstSpirit servers running in Windows environments.

9.2 Discontinued functions in FirstSpirit version 5.2

The following functions are no longer available in FirstSpirit version 5.2 and higher:

- **Snappy compression algorithm:** The "Snappy" compression algorithm for communication between the FirstSpirit clients and server that was available in versions of FirstSpirit up to and including 5.1 has been removed in FirstSpirit version 5.2. You used to be able to select this via the following locations:
    - ServerManager / Project properties / Repository
    - Start page / Connection settings / Compression
    - ServerManager / Server properties / WebStart
    - ServerManager / Server properties / Start page

  If you selected the "Snappy" compression algorithm before updating to FirstSpirit version 5.2, the "No compression" or "Deflate speed" setting will be applied automatically in version 5.2. If you used "Snappy" to compress repositories in earlier versions of FirstSpirit, you will still be able to use them in FirstSpirit version 5.2. The default compression setting for newly installed version 5.2 servers is "No compression".

  In addition "Snappy" has now been deprecated as a value for the `ConnectionManager` class (`de.espirit.firstspirit.access` package, FirstSpirit Access API).
- **"Mozilla Firefox" browser engine:** Over the course of FirstSpirit version series 5.2, support is to be withdrawn for "Mozilla Firefox" as a browser engine for the integrated preview in FirstSpirit SiteArchitect ("View / Browser engine / Mozilla Firefox (v3) (outdated)" menu and "View / Browser engine / Mozilla Firefox (v15)" menu). As an alternative, you can use the "Google Chrome" or "Internet Explorer" browser engines, which are being officially released as part of FirstSpirit version 5.2.
- **API deprecations:** Functions of the **FirstSpirit Access API** that were discontinued in FirstSpirit version 4.2 or earlier ("deprecations"), as well as functions of the **FirstSpirit Developer API** that were discontinued in FirstSpirit version 5.1 or earlier, may have been withdrawn in FirstSpirit version 5.2. For information about API changes in FirstSpirit version 5.2, see
    - chapter 6.1, page 64
    - FirstSpirit API documentations

- *FirstSpirit Community*, "Developers / Blog"[9]

9.3 Notices for future versions

- **Discontinuation of FirstSpirit modules:** The following modules are due to be withdrawn/replaced as part of FirstSpirit version 6.0:
  - **FirstSpirit DynamicDatabaseAccess:** This module for connecting various database technologies is to be replaced by the "UX-Bridge" module.
  - **FirstSpirit EnterpriseSearch:** This module for integrating the EXALEAD CLOUDVIEW™ enterprise search engine technology is to be replaced by the "GSA-Connect" module (manufacturer: TWT).
  - **FirstSpirit TranslationConnect:** This module for the structured export and import of translatable content from FirstSpirit is to be replaced by "TranslationStudio" (manufacturer: I-D Media AG).
  - **FirstSpirit BasicSearch:** This module for connecting various search engines to FirstSpirit may be replaced by a completely new implementation.

  For more information on the modules, see also http://www.e-spirit.com/marketplace/overview and the FirstSpirit Online Documentation, chapter "Documentation / Additional documentation".
- **Discontinuation of FS_LIST:** As part of FirstSpirit version 5.2, two new input components have been introduced which support functions for bundling content and thus take over most of the functions supported by FS_LIST:
  - FS_CATALOG
  - FS_INDEX

  The current FS_LIST implementation is being maintained for the time being for reasons of compatibility. However, its discontinuation (deprecation) is being planned to coincide with FirstSpirit version 6.0. It will be completely removed in a later version.
- **Discontinuation of outdated rule syntax:** Certain tags that could be used by templates on the "Rules" tab have been deprecated with the release of FirstSpirit version 5.2. These will be withdrawn completely as part of a future version. See chapter 6.3, page 71.

[9] https://community.e-spirit.com/community/developer/blog/2015/06/26/api-diff-overview-for-firstspirit-52

- **Discontinuation of auto-release mode:** As part of FirstSpirit version 6.0, support is to be withdrawn for those projects that do not work with releases (FirstSpirit ServerManager, project properties, "Options" area, "Use release function" option disabled).
- **API deprecations:** Functions of the **FirstSpirit Access API** that are being discontinued in FirstSpirit version 5.2 ("deprecations") may be withdrawn in FirstSpirit version 6.0. Functions of the **FirstSpirit Developer API** that are being discontinued in FirstSpirit version 5.2 may be withdrawn in a future minor version. API functions set to the "deprecated" state can still be used, but as they will be omitted in subsequent versions, this is not advisable. For information about API changes in FirstSpirit version 5.2, see
    - Chapter 6.1, page 64
    - FirstSpirit API documentations
    - *FirstSpirit Community*, "Developers / Blog"[10]

[10] https://community.e-spirit.com/community/developer/blog/2015/06/26/api-diff-overview-for-firstspirit-52