



FirstSpirit™

Unlock Your Content

FirstSpirit™ CorporateMedia FirstSpirit™ Version 5.1

Version	1.6
Status	RELEASED
Date	2015-02-04
Department	FS-Core
Copyright	2015 e-Spirit AG
File name	CMED51EN_FirstSpirit_CorporateMedia

e-Spirit AG
Stockholmer Allee 24
44269 Dortmund | Germany

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

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

e-Spirit

Table of contents

1	Introduction.....	3
1.1	Objective	3
1.2	Topic of this documentation	4
1.3	Terms and concepts.....	4
1.3.1	Target and remote project.....	4
1.3.2	Permission assignment.....	5
1.3.3	Generation	5
1.3.4	Release.....	5
2	Configuring remote projects.....	6
2.1	Configuring existing remote projects.....	7
2.2	Configuring a new remote project	8
2.3	Defining a technical user for a project.....	10
2.4	Defining permissions.....	11
2.4.1	Releasing objects for access	12
2.4.2	Protecting objects from access	13
2.4.3	Creating objects in a remote project.....	14
2.4.4	Releasing objects in a remote project	15
3	CorporateMedia for developers	16
3.1	Access to remote projects	16
3.2	<PROJECTS> tag.....	18
3.3	FS_REFERENCE input component	19
3.4	Links via generic link editors	22



3.5	Expanding the Access API (GetRemoteUserService)	22
3.6	Output of references via \$CMS_REF(...)\$.	23
3.7	Output of references via \$CMS_VALUE(ref:...)\$	24
3.8	Outputting remote links	24
3.9	Extension for PDF links	25
3.10	Resolutions in remote projects	25
3.11	Support for multi-language media	28
4	CorporateMedia for editors	31
4.1	Selecting objects	31
4.2	Creating objects	33
4.2.1	Media language dependence	35
4.3	Inputting links	35
4.3.1	Selecting an image	36
4.4	Reference graph visualization	37
5	Legal notices	38



1 Introduction



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 Objective

The objective of the "CorporateMedia" concept is to create all media in a separate media project and to manage the files centrally there. Then, all involved FirstSpirit projects can access the media inventory (images and files) using "FirstSpirit CorporateMedia".

In contrast to distributing media using "FirstSpirit CorporateContent"¹, the media do not have to be imported into the involved projects, but instead they can be referenced directly using CorporateMedia access. The objects physically remain in the media project in the process, but they can be used in any desired project.

The advantages of "CorporateMedia" functionality are:

- No additional memory requirement for media used in multiple projects.
- Simplified updating and managing since all media are located in a centralized media project.
- Shortened generation time for participating target projects.

¹ For further information refer to the documentation for FirstSpirit™ CorporateContent





„CorporateMedia" is supported in ContentCreator as well. This means objects (e.g. pictures) can be selected from remote projects or uploaded into a remote project. Following links which refer to a remote project is not supported either in ContentCreator or in SiteArchitect's integrated preview.

The following chapters describe concepts and configuration options for FirstSpirit™ "CorporateMedia" functionality. If a valid license for "FirstSpirit CorporateMedia" is present, the function can be enabled using FirstSpirit™ ServerManager (refer to Chapter 2, page 6).

1.2 Topic of this documentation

The documentation is divided into user groups:

- Documentation for administrators in Chapter 2.
The chapter describes the *configuration settings* for remote projects on the server and goes into detail regarding *permission assignment* for technical users (starting on page 6).
- Documentation for *developers* in Chapter 3.
The chapter illustrates some important *aspects for developers* and talks about the extensions to *input components*, the *Access-API* and *template notation* in particular (starting on page 16).
- Documentation for *editors* in Chapter 4.
The chapter covers all of the functions for *uploading and evaluating* external media objects in target projects. In addition, "*reference graph visualization*" is explained in relation to remote objects (starting on page 31).

1.3 Terms and concepts

1.3.1 Target and remote project

"CorporateMedia" allows access from one project (the "target project") to other projects (called "remote projects") located on the same FirstSpirit server. This way objects (such as media) managed in a central project can be used by different projects (also called "target projects"). The objects are only referenced in the process; they physically remain in the remote project.



1.3.2 Permission assignment

Permission assignment is controlled for the "CorporateMedia" functionality using technical users. A technical user is defined using FirstSpirit ServerManager (see Chapter 2.2, page 8) and controls access from one of the target projects to the stores for the remote project. The user requires access permissions within the remote project for this. Different access permissions to a CorporateMedia project are configured using remote project configurations in the target project (see Figure2-3, page 8). Multiple technical users can be defined in the process; these users access the same remote project, each with different access permissions (refer to Chapter 2.1, page 7 for more information).



The group "Everyone" has to be granted "Read" permission to the corresponding remote project's media so that remote media can be displayed correctly in the preview for a technical user.

1.3.3 Generation

The media from the remote projects are not taken over in the target projects, either during generation or during previewing. Instead, special link references are created. The remote media are not located, as is usual in other cases, in the project's generation directory. Therefore, the FirstSpirit deployment mechanisms that ensure that all objects being used are also published on the live system do not take effect for remote media.

(A server-side script that reacts to changes in the remote project could be used, for instance, to ensure that all of the remote media in use have been deployed at the time of publication.)

1.3.4 Release

The release configuration of projects is also applied to CorporateMedia objects. If the project for a remote project is one that uses releases, then the objects referenced in the target project have to be released in the remote project. If media are loaded into the remote project using the upload function of the respective input component, they are not released automatically (if the remote project does not use an automated release). However, it is possible to configure an automated release of these CorporateMedia objects directly in the target project. The `autoReleaseAfterUpload` parameter allows the release of the media file in the remote project using the input components of the target project. The technical user has to have the required permissions in the remote project for releasing an object (see Chapter 2.4.4, page 15).



2 Configuring remote projects

Target projects are configured in ServerManager.

Lizenz

```
license.ID=  
#FIRSTspirit license  
#Thu Jan 02 14:46:16 CET 2014  
license.USER=e-spirit  
license.EXPDATE=20.07.2014  
license.MAXPROJECTS=0  
license.MAXSESSIONS=0  
license.MAXUSER=0  
license.SOCKET_PORT=0  
license.VERSION=5  
license.MODULES=personalisation, search, integration, newsletter, portal  
license.WEBEDIT=1  
license.WORKFLOW=1  
license.REMOTEPROJECT=1  
license.PACKAGEPOOL=1  
license.DOCUMENTGROUP=1  
license.ACCESS_API=1  
license.APPTAB_SLOTS=20  
license.ARCHIVE=0  
license.CLUSTERING=1  
license.ENTERPRISE_BACKUP=1  
license.HIGHAVAILABILITY=1  
license.OFFICE_IMPORT=1  
license.OFFICE_INTEGRATION=1  
license.SCOPE=CORPORATE  
license.TYPE=PRODUCTION
```

Figure2-1: License entry for configuring remote access

To the extent that a valid license for accessing remote projects exists, the properties for remote projects and technical users can be configured under the menu item "Remote projects" (see Figure2-2).

- Configuring existing remote projects (see Chapter 2.1, page 7)
- Configuring a new remote project (see Chapter 2.2, page 8)
- Defining technical users (see Chapter 2.3, page 10)
- Defining permissions (see Chapter 2.4, page 11)



2.1 Configuring existing remote projects

One or more remote project can be configured for each target project. The selected remote projects are displayed in the project properties (for the target project).

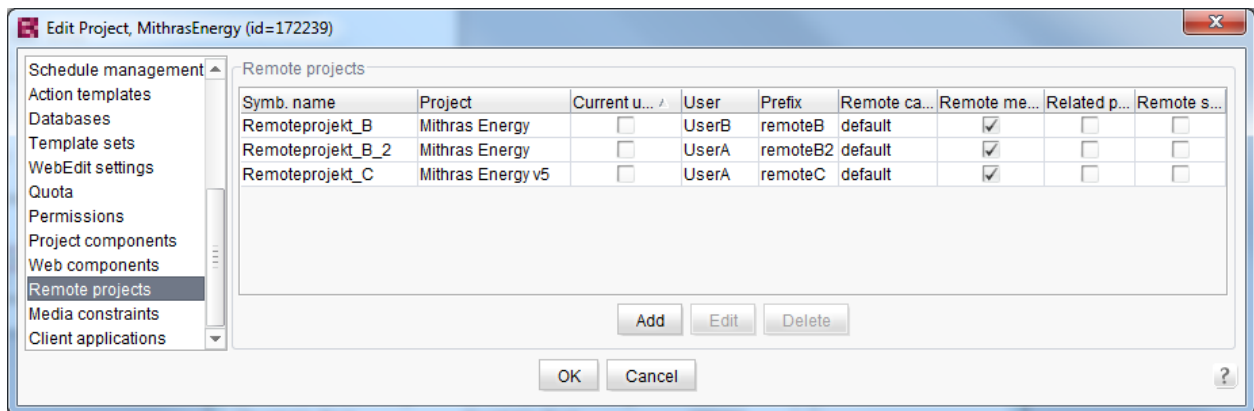
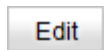


Figure2-2: Remote projects (project properties)

Three remote project configurations are present for two remote projects (column: "projects") in the figure. The target project receives access to these projects and can access the stores for the projects, reference the objects available there and upload media from the target project to the stores of the remote projects (depending on the technical user's permissions).




Clicking on "Edit" or double-clicking the table entry opens the "Edit remote project configuration" window (see Figure2-3). The configuration for an existing remote project can be modified in this window.



Clicking on "Delete" removes the configuration for the existing remote project.



2.2 Configuring a new remote project

 Clicking on the button opens the "Edit remote project configuration" dialog box. The configuration for a new remote project can be edited in this window.

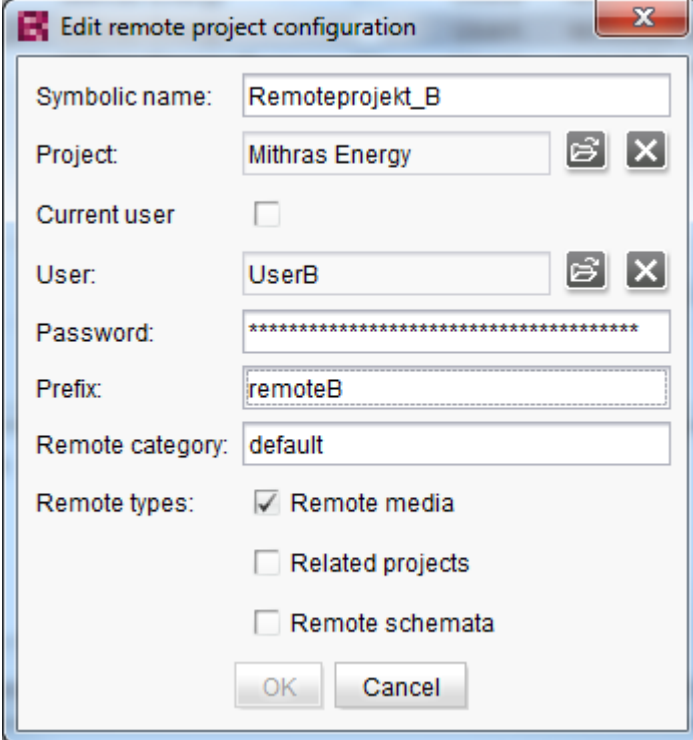


Figure2-3: Configuring a remote project


Symbolic name: Unique name for the remote project. The specific remote project can be referenced in the target project using this name by specifying the symbolic name of the remote project in the template for the "remote" attribute. For the example from Figure2-3, the specified remote project can be accessed using `remote="Remoteprojekt_B"` (see Chapter 3, page 16).


As many different symbolic names as needed can be specified for each project. The project is then displayed more than once in the "Remote projects" table; each time with another symbolic name (and, used logically, with another technical user each time) (see Chapter 2.3, page 10):




Remote projects								
Symb. name	Project	Current ... /	User	Prefix	Remote c...	Remote m...	Related ...	Remote ...
Remoteprojekt_B	Mithras Energy	<input type="checkbox"/>	UserB	remoteB	default	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Remoteprojekt_B_2	Mithras Energy	<input type="checkbox"/>	UserA	remote...	default	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Remoteprojekt_C	Mithras Energy v5	<input type="checkbox"/>	UserA	remoteC	default	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 2-4: A remote project with different technical users

 Only projects that are on a server with the target project can be configured as remote projects.

 The resolutions of remote and target projects can differ. These different resolutions have to be taken into account when developing the template (see Chapter 3.10, page 25).

Project: Clicking on the  "Select project" icon opens the dialog box for selecting the remote project:

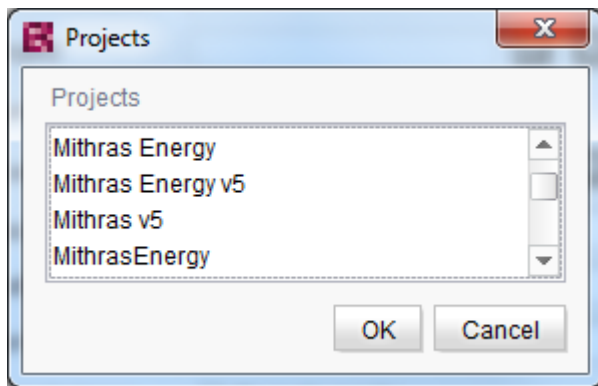


Figure 2-5: Selecting a remote project for a target project

All of the projects available on the server are displayed in the list. The selected remote project is the project that media and files are referenced from and where new media and files can be loaded to.

User: Login name for the "technical user". The technical user controls access to the respective remote project.

Password: Password for the "technical user".



Prefix: The prefix is required for generating the URL when generating the target project.



The technical user has to be known as a user in the target project and in the remote project and, in the individual stores for the remote project, has to have at least the "Visible" permission for the respective store (see Chapter 2.4, page 11).

Remote category: A category designation for a remote project configuration is specified in this field. This can be used to combine multiple remote projects into one group. The category can be used in a template for being able to select from the defined group of remote projects. By default, the field is prepopulated with the value "default".

Remote types: Three different types of remote access are possible:

- Remote media
- Related projects (see FirstSpirit manual for administrators)
- Remote schemes (see documentation for the FirstSpirit integration module)



All of the types of remote access involve additional functions that are license-dependent (see Figure2-1).

2.3 Defining a technical user for a project

Technical users are defined using FirstSpirit Server- and Project Configuration. One or more technical users can be defined in the project properties for the target project for each remote project being used (see Figure 2-4).

The technical users have to be known as users in the respective remote projects and, in the individual stores, have to have at least the "Visible" permission at the root node of the respective store (for permission assignment refer to Chapter 2.4, page 11).

Different symbolic names can be specified for each remote project. This makes it possible to assign multiple different technical users to one remote project. This configuration makes sense whenever a target project needs users with different access permissions.

For instance:

- Technical_User_A: Can reference media in the remote project's entire media store.
- Technical_User_B: Can load media into a defined folder in the remote project.



- Technical_User_C: Can only reference media from a defined folder in the remote project's media store.

2.4 Defining permissions

The permissions for technical users are specified in the remote project (SiteArchitect). The permissions for a user can be defined via the context menu using the "Tools / Change permissions" menu item. This opens the dialog box for permission assignment:

Permissions defined in this object												
User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release	Show metadata	Change metadata	Change permissions
Administrators	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Chief Editor	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Everyone	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Editor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UserB (UserB)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 2-6: Permission assignment for the technical user in the remote project

If the context menu at the root node is called up, for example, the permissions are passed down throughout the entire media store. If the context menu is called up at a lower-level object of the media store, it is not specified for the entire media store, but rather just for that object and all subordinate objects.

The technical user (here: "User B" – see Figure 2-6) has to have at least the "Visible" permission within the one remote project store. Otherwise access to the respective remote project store is not provided.



The group "Everyone" has to be granted "Read" permission to the corresponding remote project's media so that remote media can be displayed correctly in the preview for a technical user.



Five permissions are relevant for "CorporateMedia":

- **Visible:** The objects (not their editorial contents) are visible in the target project and can be referenced. If a technical user has only been granted "Visible" permission, no image preview is displayed for the remote media, for instance (see Chapter 2.4.1, page 12).
- **Read:** Only used in combination with "Visible" permission. In addition to the selection option in the project tree (via "Visible" permission), the editorial contents of the object (such as an object preview) can be displayed (see Chapter 2.4.1, page 12).
- **Create object:** New objects can be created. If this permission is set for a media store folder, separate media can be created in this folder (see Chapter 2.4.3, page 14).
- **Change:** Only needed in combination with "Create object" permission (see Chapter 2.4.3, page 14).
- **Release:** Only needed in combination with "Create object" and "Change" permissions. If this permission is set for a media store folder, media that have been newly created in the remote project via the target project can be released directly (see Chapter 2.4.4, page 15).

2.4.1 Releasing objects for access

Permissions defined in this object

User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release	Show metadata	Change metadata	Change permissions
Administrators	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Chief Editor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Editor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Everyone	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UserA (UserA)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 2-7: Permission assignment - Enabling access

In order to release objects from the remote project's media store for access from the target project, the technical user has to have at least "Visible" permission for the object. Access to a lower-level object (below the root node) is then only possible if all of the elements of the parent chain also possess "Visible" permission.

If the technical user also has "Read" permission, the editorial contents of the object (such as an image preview) are displayed in the selection dialog boxes (e.g. media selection) and within the input components. All subordinate objects inherit these rights if they are granted directly to a store's root node. This makes the corresponding remote project store (including editorial content) completely visible in the target project and the media and files available there can be referenced



from the target project.

If only individual objects in a store are to be released for access, "Visible" permission has to be granted at the root node of the corresponding store so that the technical user receives access to that remote store. Then the permissions for the technical user have to be withdrawn again ("No permissions") for all subordinate objects. Only the specified folder or folders (and parent nodes) then receive "Visible" permission (or other permissions). This way, only this folder and all subordinate objects can be referenced in the target project. If individual media files in the folder are to be protected from access, permissions have to be explicitly disabled on those objects.

2.4.2 Protecting objects from access

Permissions defined in this object												
User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release	Show metadata	Change metadata	Change permissions
Administrators	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Chief Editor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Editor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Everyone	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UserA (UserA)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UserB (UserB)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 2-8: Permission assignment – Prohibiting access

If "Visible and "Read" permissions (for a technical user) are disabled for a store's folder, that folder and all subordinate objects and objects in the target project are not visible. Objects can be blocked from access via the target project this way.

In the example from Figure 2-8, access is denied for technical user "User B" by withdrawing permissions. Access to the object for technical user "User A" remains intact, however.

Permissions can also be disabled for individual images or files as a matter of course. These objects can then no longer be selected or referenced in the target project.

In order to disable the permissions for individual folders and objects, the context menu for these objects is called up and the "Suspend permission inheritance hierarchy" checkbox is disabled in the "Permission assignment" dialog box. Permissions can then be set from scratch.



2.4.3 Creating objects in a remote project

Permissions defined in this object

User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release	Show metadata	Change metadata	Change permissions
Administrators	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Chief Editor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Editor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Everyone	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UserA (UserA)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 2-9: Permission assignment – Uploading objects

To not only reference objects from the remote project but to be able to create objects in the remote project as well, the technical user also needs "Change" and "Create object" permissions. These permissions apply to an entire store if they are enabled at the remote store's root node. However, it is also possible to release a specific upload folder for uploading. In this case, "Change" and "Create object" permissions are not granted at the root node but at that folder instead. The "Suspend permission inheritance hierarchy" checkbox is enabled for this folder and then the "Change" and "Create object" permissions are set for the technical user to accomplish this.



The input component has to be configured accordingly for uploading media using the `upload="yes"` attribute.



2.4.4 Releasing objects in a remote project

Permissions defined in this object

User Group	No Permissions	Visible	Read	Change	Create object	Create folder	Remove object	Remove folder	Release
Administrators	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Chief Editor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Editor	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Everyone	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
UserA (UserA)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 2-10: Permission assignment – Releasing objects

If objects can be created in a remote project using an input component's upload function, they are not released automatically (if the remote project does not use an automated release). Automatic release of newly created CorporateMedia objects is possible using the `FS_REFERENCE` input component in the target project. The technical user requires "Release" permission for this, in addition to "Change" and "Create object" permissions required for uploading (see Chapter 2.4.3, page 14).



The input component has to be configured accordingly for releasing media using the `autoReleaseAfterUpload="yes"` attribute.



3 CorporateMedia for developers

3.1 Access to remote projects

Remote projects are accessed in FirstSpirit SiteArchitect using the FS_REFERENCE FirstSpirit input component. This supports the selection of CorporateMedia objects (in the target project) and the creation and release of new CorporateMedia objects in the remote project.

In addition, references to the remote project can be defined using generic links. Link templates are used for configuration in this case (see Chapter 3.4, page 22).

Depending on the application, components can be configured in the form so that it is possible to select

- Local media (images and files)
- Remote media (images and files)
- Local and remote media (images and files)

The media is displayed in a new dialog for selecting media where any number of stores (local and any number of remote) can be displayed depending on the configuration. The stores are also marked with the project name in the process (see Figure 3-1).



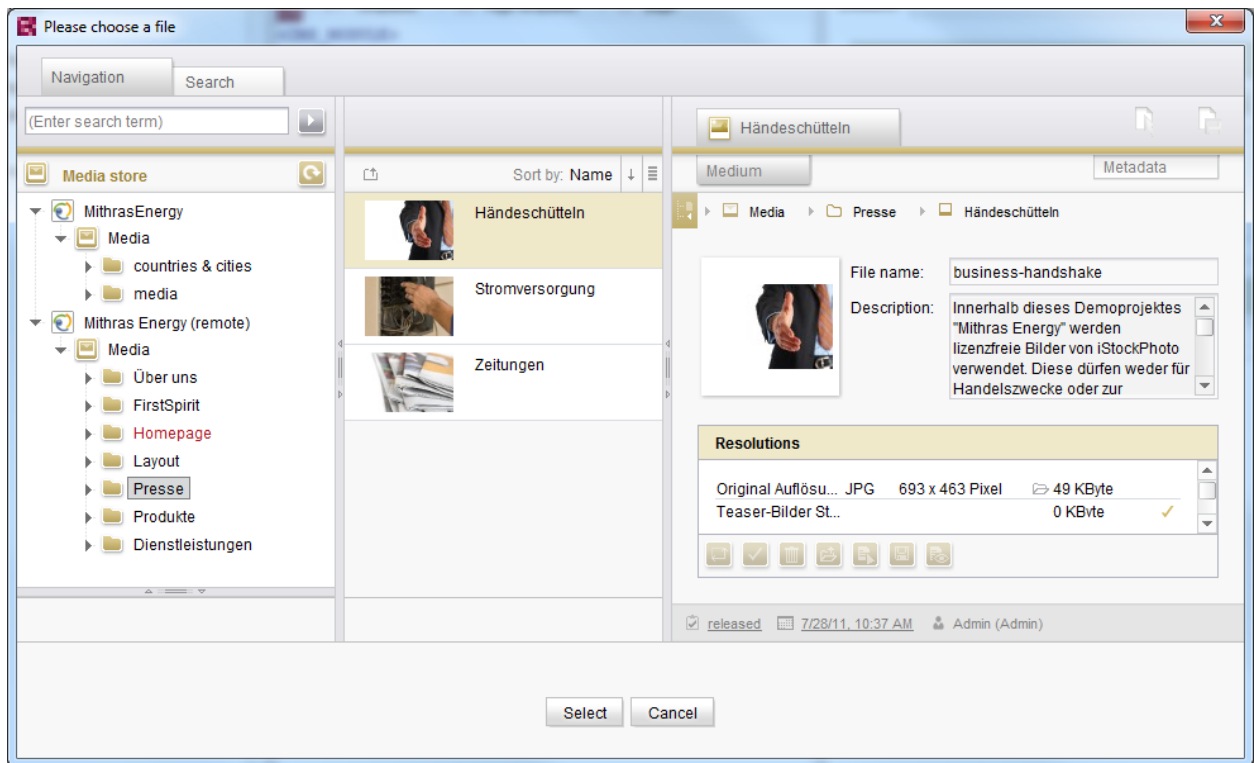


Figure 3-1: Media selection with local and remote media store



3.2 <PROJECTS> tag

Access to remote projects is configured for the FS_REFERENCE input component in the form area using the <PROJECTS> tag. It is possible to specify the following tags in this project definition:

- **LOCAL:** This tag can only be set once per project definition and allows an upload folder to be defined in the local target project.
In the process, the local project is specified by default using the `name` parameter via `<LOCAL name="."/ >`. The `name` parameter cannot be empty.
Exactly one upload folder can be specified for uploading files and media to the local project using the `uploadFolder` parameter.
- **REMOTE:** This tag can be defined multiple times, i.e. multiple remote access instances can be configured for one input component. The symbolic project name for the remote project is specified using the `name` parameter in the process.
Exactly one upload folder for uploading files and media to a remote project can be specified using the `uploadFolder` parameter.
The `autoReleaseAfterUpload` parameter controls the automatic release of media after uploading them to the remote project.
- **CATEGORY:** This tag can be defined multiple times.
The name of a remote category is specified using the `name` parameter in the process.



Only the first definition is used if two or more REMOTE or CATEGORY tags are defined with the same name.



If a category already contains defined remote projects, these projects in the category are ignored.

The order of project definitions in the form area affects the order in the selection and upload dialog (the local project is exempt from this sorting).



Example configuration:

```
<PROJECTS>

  <LOCAL name="." uploadFolder="local_folder"/>

  <REMOTE name="warehouse" uploadFolder="product_pictures"/>

  <CATEGORY name="remotemedia"/>

</PROJECTS>
```

3.3 FS_REFERENCE input component

Any number of references can be integrated into a page or section using the `FS_REFERENCE` input component. The type of referenced object is not defined here: references to media (images and files), to page references (site store) and to every other store area can be selected.

The following attributes are relevant for `FS_REFERENCE` for CorporateMedia functionality:

- `upload`

using the `REMOTE` tag:

- `name`
- `autoReleaseAfterUpload`
- `uploadFolder`

using the `CATEGORY` tag:

- `name`

In addition, the selection options can be restricted for the editor to such an extent that only specific **object types** can be selected from the remote project. This happens via the specification of the `FILTER` and `ALLOW` and/or `HIDE` tags. The remote project (or projects) has to be defined using the `PROJECTS` and `REMOTE` tags so that the `FILTER`, `ALLOW` and `HIDE` tags are taken into account for that remote project:

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

  <FILTER>

    <ALLOW type="picture"/>

    <ALLOW type="file"/>

  </FILTER>

</FS_REFERENCE>
```



```
<HIDE type="pageref"/>
</FILTER>
<LANGINFOS>
    ...
</LANGINFOS>
<PROJECTS>
    <LOCAL name=".">
        <SOURCES>
            <FOLDER name="root" store="mediastore"/>
        </SOURCES>
    </LOCAL>
    <REMOTE name="Sym-Name"/>
</PROJECTS>
    ...
</FS_REFERENCE>
```

FILTER: This tag can be used to specify which object types are displayed and/or can be selected as references in the selection dialog. A HIDE or ALLOW tag has to be specified for each respective object type that is to be hidden from the selection dialog or selectable in it. All of the object types are displayed and can also be selected if FILTER is not declared.

Important: The definition of HIDE and ALLOW, however, depends on whether and which folders are specified from which stores using the SOURCES and FOLDER tags. Only object types contained in stores defined using FOLDER can be taken into account. Thus, `<ALLOW type="pageref"/>` has no effect if `<FOLDER name="root" store="mediastore"/>` is specified as a restriction for the media store, for instance. Nothing can be selected from the selection dialog if, as in this example, only page references are specified as selectable object types and the remote project's media store is specified using FOLDER. Therefore, the definition of FILTER and FOLDER should be carefully considered.

HIDE: The object types in the remote project that are hidden from the selection dialog, and thereby can no longer be selected, can be defined using the HIDE tag. A HIDE tag has to be specified for each type and this is done using the `type` parameter for each one. If HIDE is not



specified, all of the object types are displayed in the selection dialog.

ALLOW: Object types in the remote project that may be selected using the selection dialog can be defined using the ALLOW tag. An ALLOW tag has to be specified for each type and this is done using the `type` parameter for each one. If ALLOW is not specified, all of the object types can be selected in the selection dialog.

type: The object type is specified in double quotes using this attribute. The following types are available for this:

- PICTURE (for images; media store)
- FILE (for files; media store)
- MEDIAFOLDER (for folders for the media store)
- PAGEREFFOLDER (for menu levels; site store)
- PAGeref (for page references; site store)
- PAGE (for pages; page store)
- CONTENT2 (for data sources; data store)



Each object type should be used only once in the input component form, either with the ALLOW tag or the HIDE tag. If a type is specified with both tags, only the ALLOW tag is evaluated.

SOURCES: Using the SOURCES tag, it is possible to limit the selection or display of defined folders (including subfolders) for the respective remote project. SOURCES is a positive list; this means only the specified folders are allowed. In order to allow a folder, a FOLDER tag has to be specified for each.

FOLDER: A FOLDER tag has to be specified for each folder that is to be taken into account for the selection from a remote project, with each being specified using the `name` and `store` parameters.

name: This attribute is used to specify the valid name for a media store folder for the remote project from which media may be selected. A value of `root` is used to allow the entire store.

store: This attribute is used to specify the name of the media store: `mediastore`.



3.4 Links via generic link editors

Link templates can, as with page and section templates, be configured by adding input components into the form area. This allows template developers to utilize all FirstSpirit input components when creating links.

Therefore, the FS_REFERENCE input component described in Chapter 3.3 can be used to utilize the CorporateMedia functionality for links (e.g. to add a link to the page of a remote project).

3.5 Expanding the Access API (GetRemoteUserService)

The remote project is accessed via the Access API using the UserService interface. The UserService interface has been expanded by adding the `getRemoteUserService(String symbolicName)` method, which returns the remote project's UserService.

The symbolic name of the remote project configuration is passed as a transfer parameter (see Chapter 2.2, page 8).

The connections for the remote project are maintained in the user's connection and they are closed when `disconnect()` is called for the user connection.

Call for RemoteUserService using the Beanshell console in the target project:

```
bsh %  
e.getProject().getUserService().getRemoteUserService("sym-Name")
```



3.6 Output of references via `$CMS_REF(...)$`

Using the instruction `$CMS_REF(...)$` initiates a reference to a path. If a remote media file is selected using an input component and the "Reference" parameter is the variable name for this component, a reference to the selected media object is created automatically:

```
$CMS_REF("st_picture")$
```

During generation, the URL is created from the media file's absolute path in the target project and appended with the URL prefix from the remote project configuration.

Example – Generation:

```
/prefix_remoteProject/media/mediafolderName/imagename_thumbnail.jpg
```

During a preview, the URL points to target project's preview directory. The absolute path at this location results from the remote project's ID and the media file's ID.

Example – Preview:

```
/fs4preview/jump?url=%2Ffs4preview%2Fpreview%2F4235451%2Fmedia%2FDE%2Fcurrent%2F4309774%2Fthumbnail%2Fx.jpg
```



Preview security is not supported for CorporateMedia objects.

The instruction `$CMS_REF(...)$` has to be expanded by adding the parameter `remote` for manual references that are not set by the editor using an input component but are defined in the templates instead. The symbolic name of the remote project is passed to the parameter:

```
$CMS_REF(media:"suedsee",remote:"sym-Name")$
```

This notation is used in link templates as well (see Chapter 3.8, page 24).

Local URLs are generated if the parameter is empty or not present.

Additional parameters, such as for a specific resolution or a specific project language, can be passed in the instruction.



3.7 Output of references via `$CMS_VALUE(ref:...)$`

The `ref(...)` function offers the option of applying methods directly to the specified media file or a page reference. This is particularly useful if a design element is intended to be specified manually in a page template, for instance.

Therefore, a page or media reference can be output from a remote project using instruction `$CMS_VALUE(ref:...)$` as well:

```
$CMS_VALUE(ref(st_picture,res:"thumbnail"))$
```

Outputting the height of a remote media file from an image input component:

```
height= $CMS_VALUE(ref(st_pictureC,res:"thumbnail").height)$
```

Refer to the FirstSpirit online documentation for more detailed information on output using the `ref` function.

3.8 Outputting remote links

While a remote reference from input components can be resolved using conventional means via the `$CMS_REF(..)$` tag², a remote attribute also has to be specified within the instruction for resolving a link.

Outputting an internal link to a remote project:

```
<a href="$CMS_REF(#link.sitestoreref,remote:#link.sitestoreref_remote)$">$CMS_VALUE(#link.text)$</a>
```

Image output of a media reference from a remote project:

```
<img src='$CMS_REF(#link.mediaeref,remote:#link.mediaeref_remote)$
```

Refer to the FirstSpirit online documentation for outputting links in link templates using the `#link` system object.

² Refer to the FirstSpirit online documentation: `../Template development/Template syntax/Instructions/cms_ref`



3.9 Extension for PDF links

Absolute links in the generation directory are used for creating PDF files in templates using FOP: `abs:3`. Because CorporateMedia objects are not copied to the generation directory, a path that refers to the remote project's project directory is created during generation (release state).

3.10 Resolutions in remote projects

The "res" or "resolution" parameter has to be specified if a different image resolution other than the original resolution (resolution with the ORIGINAL identifier) is to be used when outputting an image reference from the (remote) media store.

A valid identifier in the project for a resolution that is also enclosed by double quotes (e.g. "thumbnail") has to be specified as the value of the parameter.

```
§CMS_REF(media:"suedsee", res:"thumbnail")§
```

Instead of a fixed, defined identifier, the "resolution" object can also be used (refer to the end of the chapter for an example).



The resolutions of remote and target projects can differ. These different resolutions have to be taken into account when developing templates. If the template developer references a remote media file with a resolution that does not exist in the remote project, this can result in errors when requesting a preview and during generation.

Identical resolutions in the target project and remote projects: In this case, no errors should occur when requesting a preview or when generating CorporateMedia objects.

Different resolutions in the target project and remote projects: In this case, errors can result if the template developer sets a fixed resolution that does not exist in the remote project in the presentation channel using the "res" or "resolution" attribute. If the editor then selects a media file from the remote media store, the resolution preset by the template developer would not be found in the remote project when generating a preview, for instance. The following message is output using the "Preview error" context menu function:



```
ERROR: invalid resolution 'thumbnail' for project 'Mithras_Remote_C' in  
'$CMS_REF(st_remotepic_remoteC, res:"thumbnail")$ at 28, 11'
```

In this case, the media file would be output in its original resolution (default value).

Resolutions with the same name in the target project and remote projects: A special case exists if different resolutions with the same name are present in the target project and remote projects.

Applicable in this case: Resolutions are only used from the project where the media file originated. If an image from a remote media store is selected, only the resolutions that have been configured for the remote project are taken into account. If, instead, an image from the local media store is selected, the resolutions from the target project are taken into account.

Example:

- Target project has a "thumbnail" resolution (width of 10 px)
- RemoteProject_B has a "thumbnail" resolution (width of 50 px)
- RemoteProject_C has a "thumbnail" resolution (width of 75 px)

Then, in the target project:

- A media file from RemoteProject_B's remote media store is output with the "thumbnail" resolution: Resolution has a width of 50 px
- A media file from RemoteProject_C's remote media store is output with the "thumbnail" resolution: Resolution has a width of 75 px
- A media file from the target project's local media store is output with the "thumbnail" resolution: Resolution has a width of 10 px



Therefore, a fixed resolution should be configured in templates only if either the resolutions in the target project and remote projects are identical or the source media store for the selected media can be known for certain.

Outputting using object resolution: Instead of specifying a fixed value using the "res"/"resolution" parameter, the template developer can also retrieve and pass the "resolution" object from the remote project via the referenced media file. Using the following instruction, the template developer could output an image from a remote project with all of the resolutions available in the remote project (example for the FS_REFERENCE input component):



```
$CMS_IF(!st_remotepic.isEmpty)$  
    $CMS_FOR(_res, st_remotepic.media.project.resolutions)$  
    <pre>  
    $CMS_VALUE(_res)$=$CMS_REF(st_remotepic, remote: st_remotepic.remoteSymName,  
    res:_res)$  
    </pre>  
    $CMS_END_FOR$  
$CMS_END_IF$
```

This instruction in a template results in the following output when previewing the local project (all of the resolutions originate from the remote project instead of the local project):

```
ORIGINAL (0 x 0)=  
/fs4preview/jump?url=%2Ffs4preview%2Fpreview%2F5372131%2Fmedia%2FEN%2Fcurrent%  
2F5372137%2FORIGINAL%2Fx.png
```

```
res1 (400 x 300)=  
/fs4preview/jump?url=%2Ffs4preview%2Fpreview%2F5372131%2Fmedia%2FEN%2Fcurrent%  
2F5372137%2Fres1%2Fx.png
```

```
res2 (1000 x 500)=  
/fs4preview/jump?url=%2Ffs4preview%2Fpreview%2F5372131%2Fmedia%2FEN%2Fcurrent%  
2F5372137%2Fres2%2Fx.png
```

```
res3 (1600 x 750)=  
/fs4preview/jump?url=%2Ffs4preview%2Fpreview%2F5372131%2Fmedia%2FEN%2Fcurrent%  
2F5372137%2Fres3%2Fx.png
```

The same instruction also works when selecting a media file from the target project's media store, in which case only the resolutions from the target project are taken into account.



3.11 Support for multi-language media

The "Corporate Media" function supports media multilingualism. This means that language-dependent remote media can be output in the target project in any of the maintained languages (of the remote project).

When specifying the optional "lang" or "language" parameter, the specified (remote) project language is also taken into account for the resolution of a reference. If the parameter is not specified, the reference is initiated with the current language (preview and generation).

A language abbreviation valid for the remote project and enclosed in double quotes is to be specified as the value (e.g. lang="ES").

```
$CMS_REF(media:"suedsee", lang:"ES")$
```

Instead of a fixed, defined identifier, the "language" object (e.g. `#global.language`) can also be used (refer to the end of the chapter for an example).



Remote and target project languages can be different. These different languages have to be taken into account when developing templates. If the template developer references a remote media file in a language that does not exist in the remote project, this can result in errors when requesting a preview and during generation.

Identical languages in the target project and remote projects: In this case, no errors should occur when requesting a preview or when generating CorporateMedia objects.

Different languages in the target project and remote projects: In this case, errors can result if the template developer sets a language that does not exist in the remote project in the presentation channel using the "lang" or "language" attribute. If the editor then selects a media file from the remote media store, the language preset by the template developer would not be found in the remote project when generating a preview, for instance. The following message is output using the "Preview error" context menu function:

```
WARN: invalid language 'FR' for project 'Mithras_Remote_C'
```

In this case, the media file would be output in the master language (of the remote project).

In contrast, it is possible to specify a remote media file in a language that is available *only* in the remote project but not in the target project. The language-dependent media file is physically located in the remote project. This allows the remote project's language-dependent media



contents to be referenced in the target project using CorporateMedia access.

Applicable in this case: Languages are only used from the project where the media file originated. If an image from a remote media store is selected, only the language-dependent media contents that have been configured for the remote project are taken into account. If, instead, an image from the local media store is selected, the language-dependent media contents from the target project are taken into account.

Example:

- Target project has German (DE) | English (EN) | French (FR) (with EN as the master language)
- Remote project has German (DE) | English (EN) | Spanish (ES) (with EN as the master language)

Then a media file from the remote media store is referenced in the target project:

- In DE: Media content for DE is output
- In EN: Media content for EN is output
- In ES: Media content for ES is output
- In FR: Media content for EN (master language) is output (error message for generation and previewing)

Accordingly, only the language-dependent media contents for DE, EN and FR (but not for ES) can be output for a media file from the target project's local media store.



Therefore, a fixed language setting should be configured in templates only if either the languages in the target project and remote projects are identical or the source media store for the selected media can be known for certain.

Outputting using object language: Instead of specifying a fixed value using the "lang"/"language" parameter, the template developer can also retrieve and pass the "language" object from the remote project. Using the following instruction, the template developer could output an image from a remote project with all of the languages available in the remote project (example for the FS_REFERENCE input component):



```
$CMS_IF(!st_remotepic.isEmpty)$  
    $CMS_FOR(_lang, st_remotepic.media.project.languages)$  
    <pre>  
    $CMS_VALUE(_lang)$=$CMS_REF(st_remotepic, remote:  
st_remotepic.remoteSymName, lang:_lang)$  
    </pre>  
    $CMS_END_FOR$  
$CMS_END_IF$
```

This instruction in a template results in the following output when previewing the local project (all of the resolutions originate from the remote project instead of the local project):

```
EN=/fs5preview/jump?url=%2Ffs5preview%2Fpreview%2F5372131%2Fmedia%2FEN%2Fcurre  
nt%2F5372137%2FORIGINAL%2Fx.png
```

```
DE=/fs5preview/jump?url=%2Ffs5preview%2Fpreview%2F5372131%2Fmedia%2FDE%2Fcurre  
nt%2F5372137%2FORIGINAL%2Fx.png
```

```
ES=/fs5preview/jump?url=%2Ffs5preview%2Fpreview%2F5372131%2Fmedia%2FES%2Fcurre  
nt%2F5372137%2FORIGINAL%2Fx.png
```

The same instruction also works when selecting a media file from the target project's media store, in which case only the languages from the target project are taken into account.



4 CorporateMedia for editors

4.1 Selecting objects

The selection of objects (images, files, page references) using the FS_REFERENCE input component is explained below:

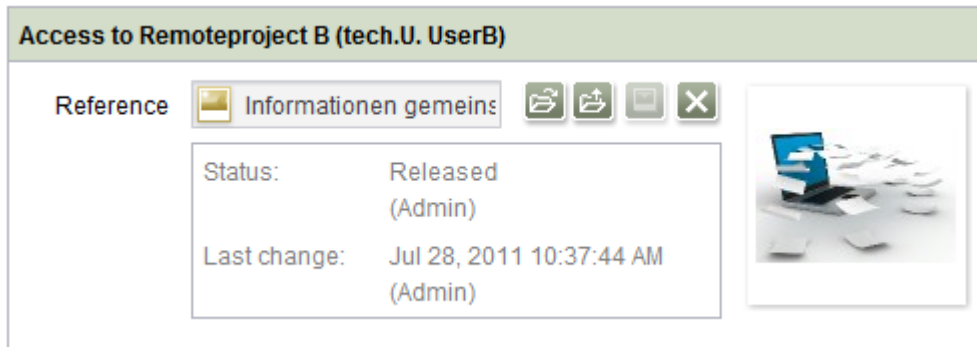


Figure4-1: Image selection using a FS_REFERENCE input component

Images, files or page references are selected from the remote project using a FS_REFERENCE input component configured accordingly (see Chapter 3.3, page 19). Depending on the configuration and access permissions, objects from the store of corresponding local and/or one or more remote projects can be referenced via this input component.



Clicking the "Remove" icon removes an already referenced object from the input component.



Clicking the "Select" icon opens the new selection dialog box for media. Depending on the configuration and access rights, these input components can be used to reference:

- Media (images and files) from the local media store and/or the media store of one or more remote projects

and/or

- Page references from the local site store and/or the site store of one or more remote projects



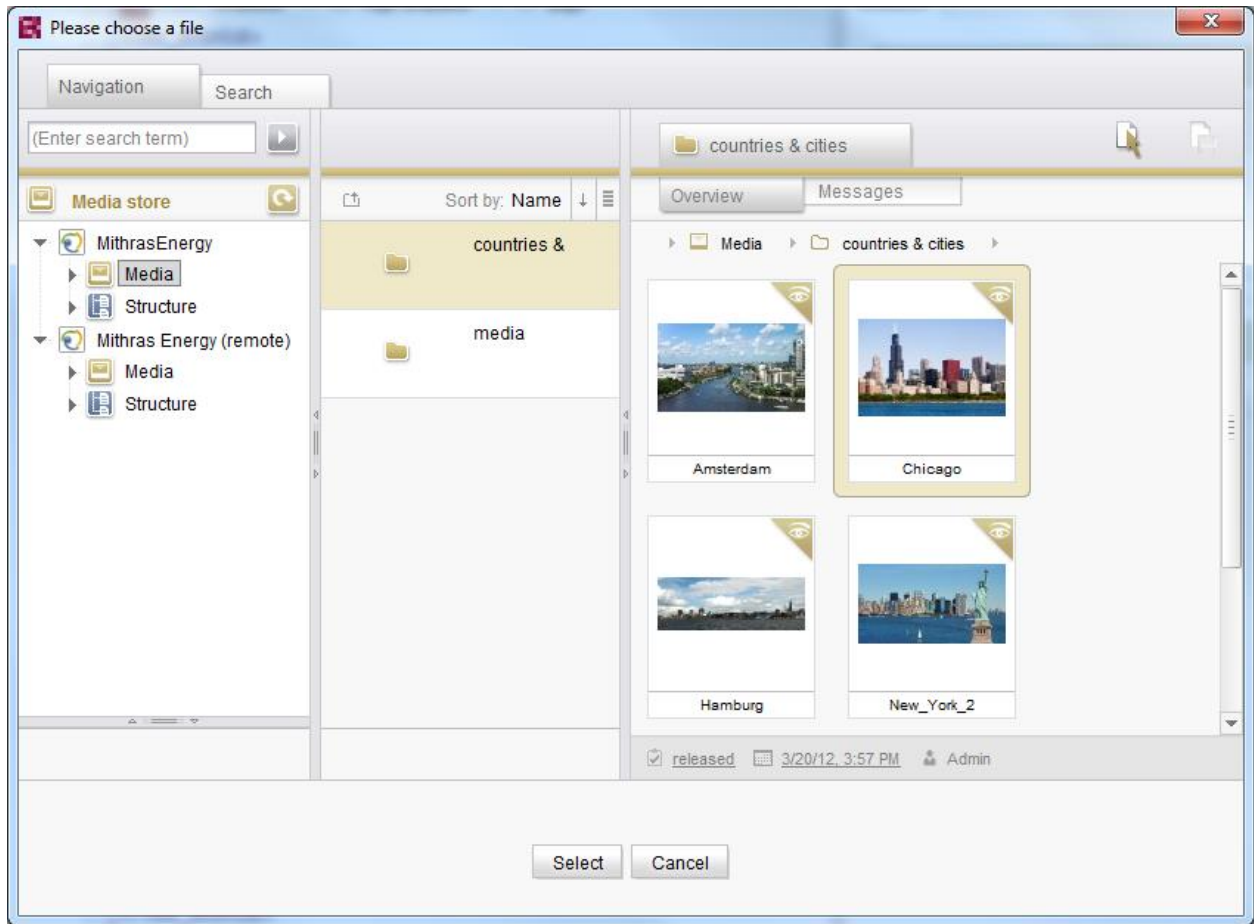



Figure 4-2: Selection from media store and site store (local and remote)

An object from the desired store can be selected in the dialog.

Cancel Clicking the button ends the operation. The object is not transferred to the input component.

Select Clicking on the button transfers the selected object to the input component and displays a preview of the object. The "Reference" field now contains a reference to the selected object.

 *If an image has been selected via the media store of a remote project, it only exists in the target project via this reference. This means that it cannot be found in the target project's media store.*





CorporateMedia does not support setting variables in the page store.



The "Parse file" function for files in the media store is not supported for CorporateMedia objects.

4.2 Creating objects

Objects are created in a remote project using a FS_REFERENCE input component configured accordingly (see Chapter 3.3, page 19). Depending on the configuration and access permissions, objects from the local file system can be uploaded a remote project's media store.



Clicking the "Upload" icon initially opens the file selection dialog in addition to the dialog box saying "Please select an upload folder". Depending on the configuration of the input component, the display includes:

- Just the project's local media store.
- Just the media store for one or more remote projects.
- Both media stores (remote and local - see Figure 4-3)



New media objects can only be created in existing folders. Directories cannot be created in the remote project's media store.





Multiple remote projects can be displayed depending on the configuration of the input component. Furthermore, uploading can be limited to a folder in these media stores.

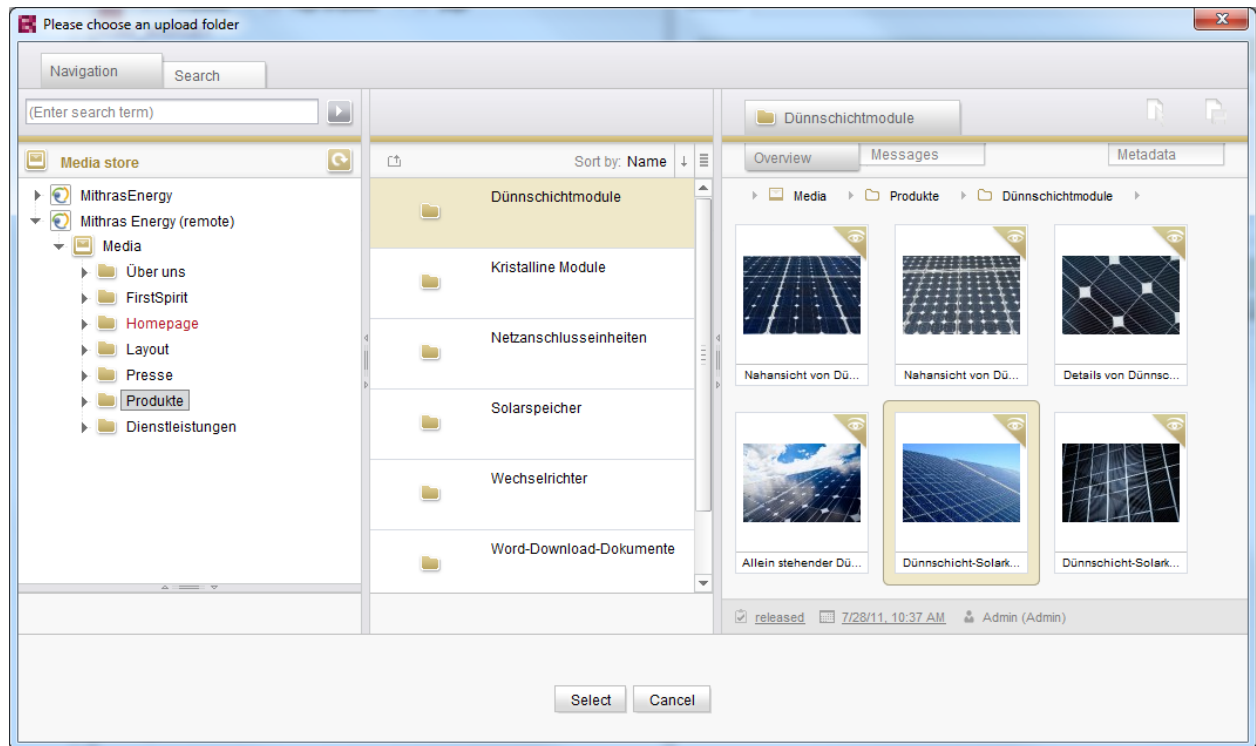


Figure 4-3: Selecting a target folder for uploading

The desired media store can be expanded in the dialog box. Then a folder is selected as the target folder for uploading.

Cancel

Clicking the button ends the operation. No upload folder is selected.

OK

Clicking on the button defines the selected folder as the target folder.



4.2.1 Media language dependence



It is no longer possible to create media in language-dependent form in a remote project when uploading in FirstSpirit version 5.0. If language-dependent media are desired, they have to be created manually in the corresponding project.

Uploading a media file using a FS_REFERENCE input component's upload function creates the file independent of language. This means, the same object is maintained in the input component for each language present in the project.

If the object has been created in a remote project and that project uses releases, the newly created object is not released automatically (see Chapter 2.4.4, page 15).

4.3 Inputting links

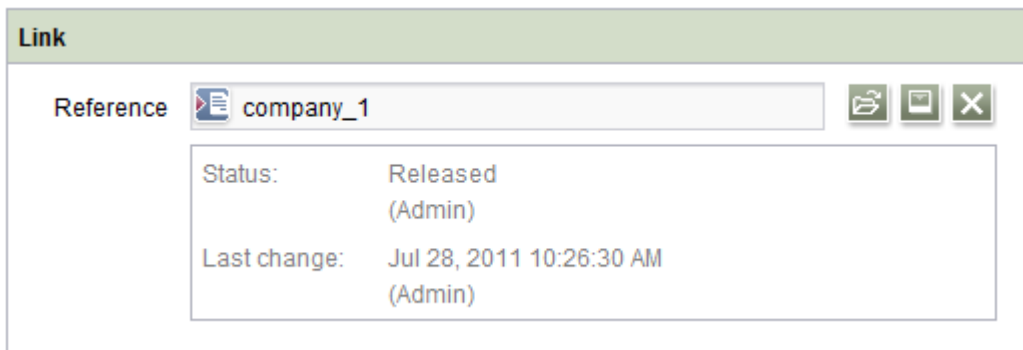


Figure 4-4: Input component for link selection

Links are created using generic link editors based on conventional FirstSpirit input components. If links to remote projects are to be embedded, the FS_REFERENCE input component that supports CorporateMedia functionality can be used. Refer to Chapter 3.1, page 16 through Chapter 3.4, page 22 in this regard.



Clicking this icon removes an existing reference from the input component.



Clicking the "Select" button in the "Target" field opens a window with a selection list for the link target. Depending on the configuration of the input component, the display includes:

- The project's local site store
- The project's local media store
- The site store for one or more remote projects



- The media store for one or more remote projects

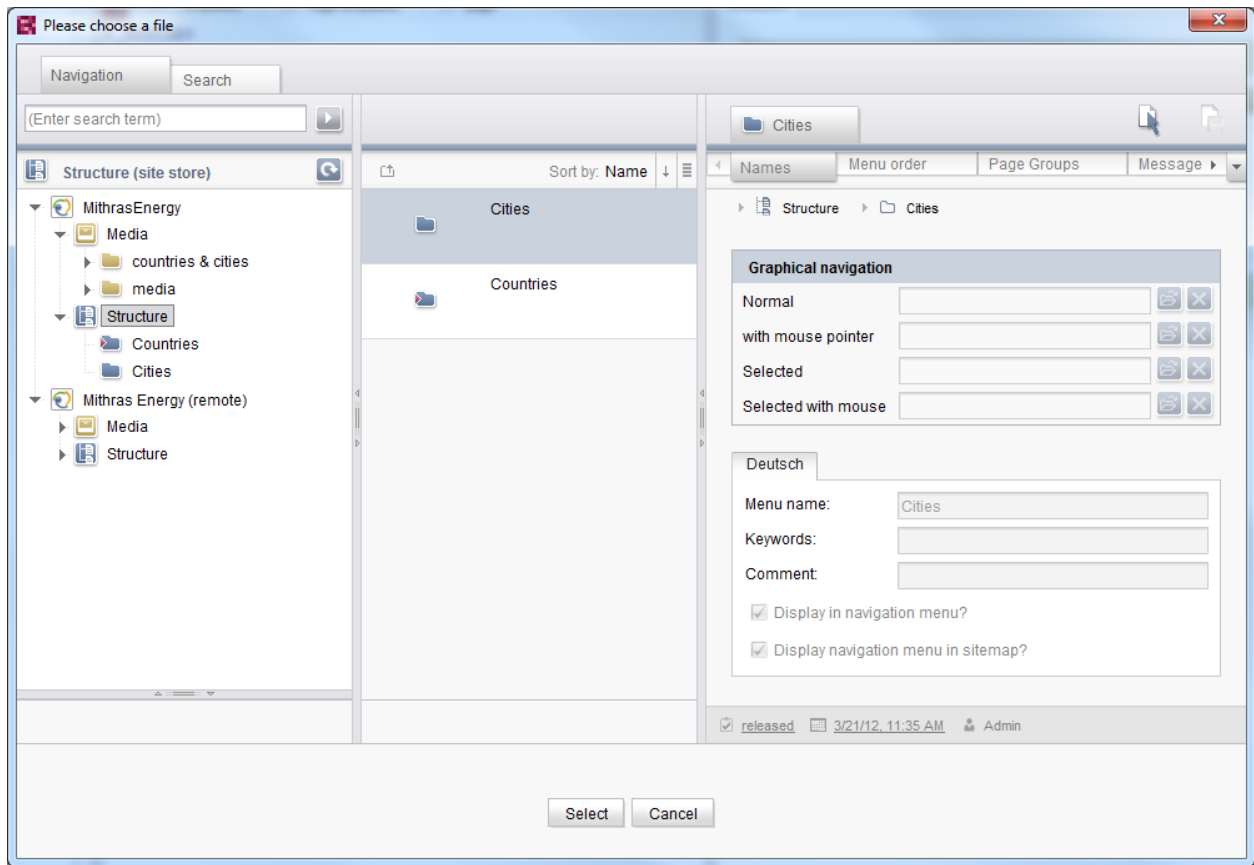


Figure 4-5: Selection option for the link target

Cancel Clicking this button cancels the operation. No link target is selected.

Select Clicking this button sets the selected media or site object as the target for the internal link. The selection is applied to the "Target" field.

4.3.1 Selecting an image

The option to select an image is available in the link input component as well. This works the same way as selecting a link target.



4.4 Reference graph visualization

Server-wide, cross-project relationships between FirstSpirit objects can be mapped using the reference graph visualization. Thus, the relationships with external objects (from a remote project) can be visualized using the reference graph as well. A distinction between the following is made in the process:

- Incoming references
(e.g. references to a media file from the remote project)
- Outgoing references
(e.g. references to a section of the target project using external media objects)

The dependencies on a remote project can be called up, such as on a page or section in the target project, using the "Tools"/"Display dependencies" context menu:

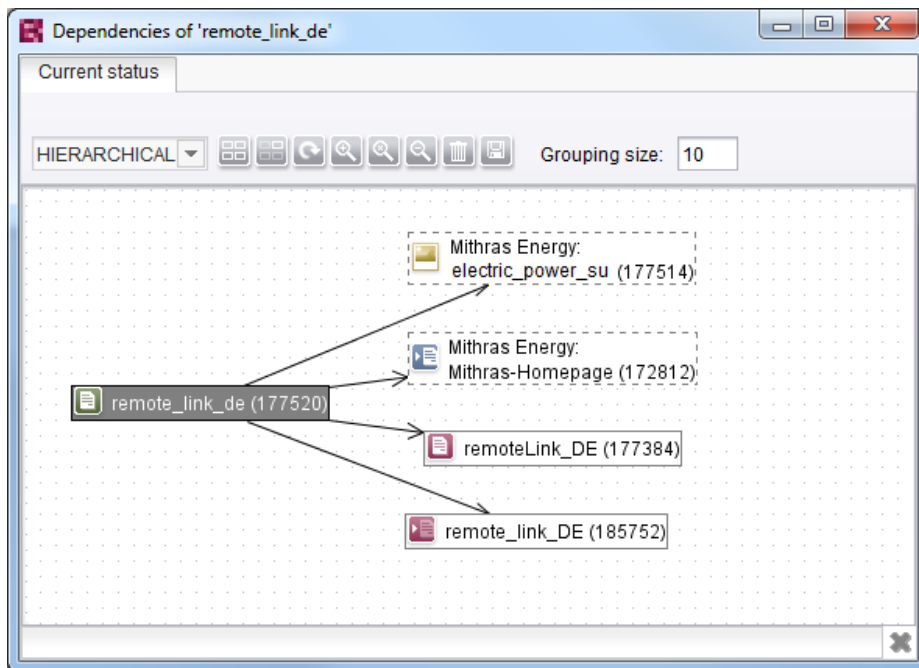
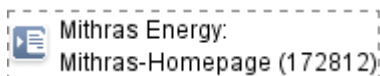


Figure 4-6: Visualization of the dependencies in the target project

Both the page reference shown in Figure 4-6 and the media object originate from a remote project. In contrast to local objects, the name of the remote project (here: "Mithras Energy") is also specified for remote references:



In the opposite case, the dependencies in remote projects, such as for a remote media file, can be visualized as well.



5 Legal notices

The "FirstSpirit™ CorporateMedia" module is a product of e-Spirit AG, Dortmund, Germany.

Only a license agreed upon with e-Spirit AG is valid with respect to the user for using the module.

Details regarding potential third-party software products in use not created by e-Spirit AG, their separate licenses and, if applicable, their update information can be found on the start page of every FirstSpirit server in the "Legal notices" area.

