

# FirstSpirit™

*Your Content Integration Platform*

## FirstSpirit™ Liferay FirstSpirit™ Version 4.x

<b>Version</b>	<b>1.0</b>
<b>State</b>	<b>RELEASED</b>
<b>Date</b>	<b>2012-07-18</b>
Department	Professional Services
Author/ Authors	M. Pahl
Copyright	2012 e-Spirit AG
File name	LIFE4xEN_FirstSpirit_Modules_Liferay

### e-Spirit AG

Barcelonaweg 14  
44269 Dortmund | Germany

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

[info@e-Spirit.com](mailto:info@e-Spirit.com)  
[www.e-Spirit.com](http://www.e-Spirit.com)

e-Spirit<sup>AG</sup>

## Table of contents

<b>1</b>	<b>Introduction.....</b>	<b>3</b>
1.1	Functional scope.....	3
1.2	Topics covered in this document.....	3
<b>2</b>	<b>Installation.....</b>	<b>4</b>
2.1	Components required.....	4
2.1.1	Java JDK.....	4
2.1.2	FirstSpirit-Server.....	4
2.1.3	Liferay account.....	4
2.2	Installing and configuring the Portal module in FirstSpirit.....	4
2.2.1	Installation.....	5
2.2.2	Configuration.....	6
2.2.3	Configure schedule.....	8
2.2.4	Deployment.....	11
2.3	Adjusting personalization.....	12
2.3.1	Creating group including roles.....	12
2.3.2	Entering group.....	14
2.3.3	Using group.....	15
<b>3</b>	<b>Extending the FirstSpirit project.....</b>	<b>17</b>
3.1	Page templates.....	17
3.1.1	Portal page.....	19
3.1.2	Mixed portal page.....	19
3.1.3	Content projection and other loose pages.....	20
3.1.4	Project settings.....	21



- 3.2 Portlet sections..... 22
  - 3.2.1 Determining portlet type ..... 22
- 3.3 Linking..... 23
- 3.4 Defining EntryPoints ..... 24
- 3.5 Handling media in the Liferay portal ..... 26
- 3.6 Metadata ..... 27
  
- 4 Differentiation between private & public community ..... 29**
  
- 5 Layout & theme..... 30**
  
- 6 Creating and subsequent deployment..... 34**
  - 6.1 Creating a page ..... 34
  - 6.2 Result in the portal ..... 37
  
- 7 Legal notices ..... 38**
  
- 8 Scripts ..... 39**
  - 8.1 moveContentXml ..... 39
  - 8.2 activePortal..... 39



# 1 Introduction

This document describes the FirstSpirit™ module for integrating FirstSpirit™ navigation structures and content, taking into account permissions assigned to different roles in the Liferay portal. In this way, an efficient, personalized and multi-application option is offered for providing content in the Liferay portal. The following explains how to install and configure the module.

## 1.1 Functional scope

The Liferay integration described here supports the following aspects:

- Provision of FirstSpirit content in the Liferay portal
- Transfer of navigations into the Liferay portal – including taking into account role definitions
- Transfer of permissions for personalized access

## 1.2 Topics covered in this document

**Chapter 2** describes the installation and project-specific configuration of the FirstSpirit Portal module.

**Chapter 3** illustrates the adjustments in the FirstSpirit project created by installation of the module and additional manual adjustments to be made

**Chapter 4** briefly describes the Liferay differentiation between the public and private community

**Chapter 5** explains the FirstSpirit adjustment for using individual layouts and themes

**Chapter 6** explains how to create and then deploy a page in the Liferay portal.



## 2 Installation

### 2.1 Components required

Before installing the Liferay module the following preparations should be made on the server:

#### 2.1.1 Java JDK

To operate the FirstSpirit Server, the Java JDK must be installed. At least version 1.5 must be installed, Java version 1.6 is recommended. The installation files can be downloaded from here: [http://java.sun.com/javase/downloads/index\\_jdk5.jsp](http://java.sun.com/javase/downloads/index_jdk5.jsp)



*For more information on the Java version please refer to the FirstSpirit Installation Guide!*

#### 2.1.2 FirstSpirit-Server

Detailed information on installing a FirstSpirit server is given in the *FirstSpirit Installation Guide*.

#### 2.1.3 Liferay account

To link a Liferay portal to FirstSpirit you need an installed and started portal and a valid account for this portal.

The installation files required are available from the site: <http://www.liferay.com/>. An account can be created following installation. The user chosen within the configuration must have all permissions within the Liferay Community.

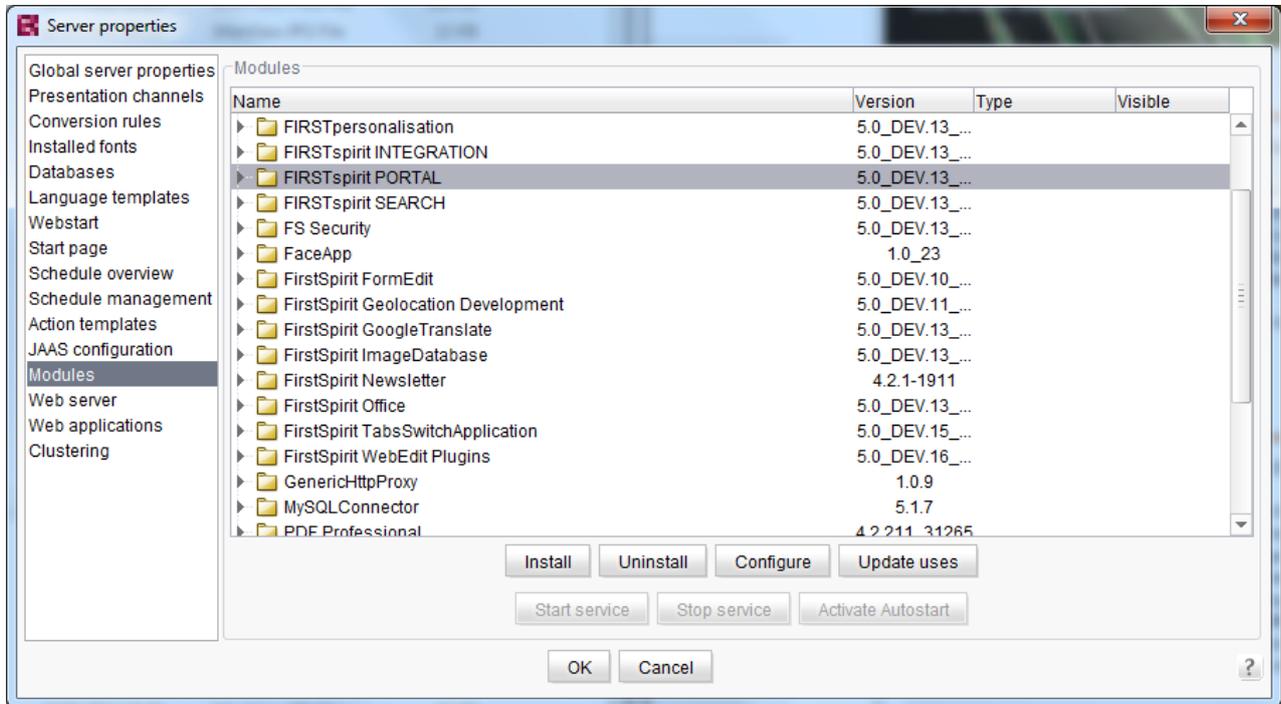
## 2.2 Installing and configuring the Portal module in FirstSpirit

To link the Liferay portal, the FirstSpirit PORTAL module is required, which is first installed on the server and then has to be added to the relevant project, before project-specific configuration can be carried out.



## 2.2.1 Installation

To install the FirstSpirit PORTAL module, open the Server and Project Configuration and select the "Modules" area of the server properties.

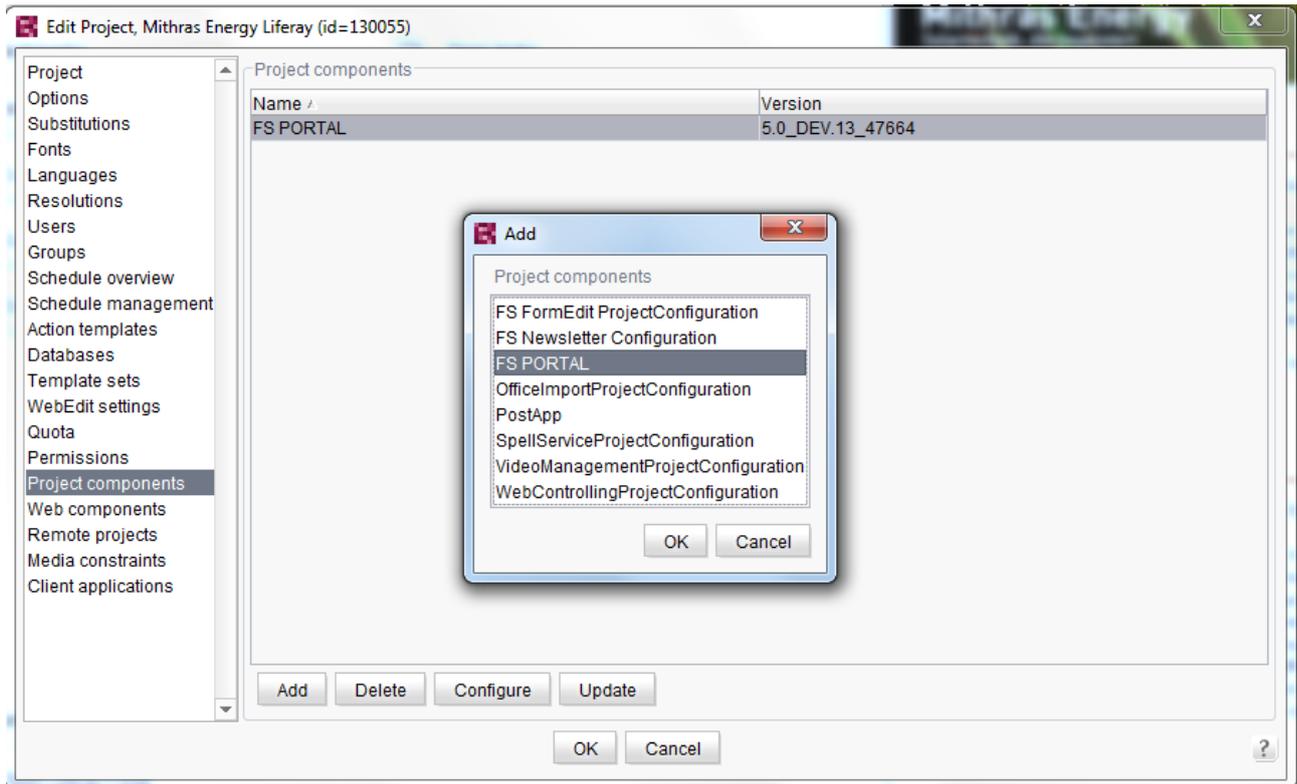


**Figure 2-1: List of the modules in the server properties**

The main panel contains a list of the modules installed on the server. After clicking "Install", select the *fs-portal.fsm* file to be installed and confirm with "Open". After the installation has been completed, a new "FirstSpirit PORTAL" folder should have been added in the module overview. Now click "OK" to close the server properties.

Before the project-specific configuration of the project to be extended to include the portal integration can be carried out, the module must be added to the project first. To do this, open the project configuration by double-clicking the relevant project in the project overview and in the left-hand menu, select the "Project Components" item. A list of all modules already assigned to the project is displayed, to which the "FS PORTAL" module can be added using the "Add" button.





**Figure 2-2: Project components in the project properties**

Following successful installation the project has been extended to include the following elements:

- **Page and section templates:** A "Portal" folder with special templates has been created in the page and section templates of the project.
- **Context menu scripts:** The "scripts" area in the Template Store has been extended to include a "Portal" folder which in turn contains a range of context menu scripts. These are used to designate selected folders in the Site Store and to add necessary portal information.
- **Action templates:** Two server-side scripts have been added to the action templates which are used to generate and activate the portal navigations.

## 2.2.2 Configuration

Double-click the "FS PORTAL" project component to open the corresponding dialog to be configured.



**Project settings**

PCD folder title

PCD folder name

Link type: NC/PCD mixed

Presentation channel: html (HTML)

Media extensions

Identifier type: Numeric IDs

UID identifier prefix

Language optimization:  Optimized xml format

**Prefix settings**

URL prefix search-xml

URL prefix navigation-xml

URL prefix internal link

Prefix for SAP KMC

**Portal settings**

Portal host

OK Cancel

**Figure 2-3: Module configuration**

The following settings are necessary for the configuration of the portal component in conjunction with the Liferay portal.

### **Project settings**

**Presentation channel:** Name of the output channel of the portal pages. This information is required for correct URL generation as the navigation only points to one output channel.

**Media extensions:** File extensions of the media which are to be included in the search (full text indexing). Several file extensions can be specified as a comma-separated list. If no value is entered the search is automatically performed through all media.

**Identifier type:** ID (numerical) must be set in Liferay, as the elements within Liferay

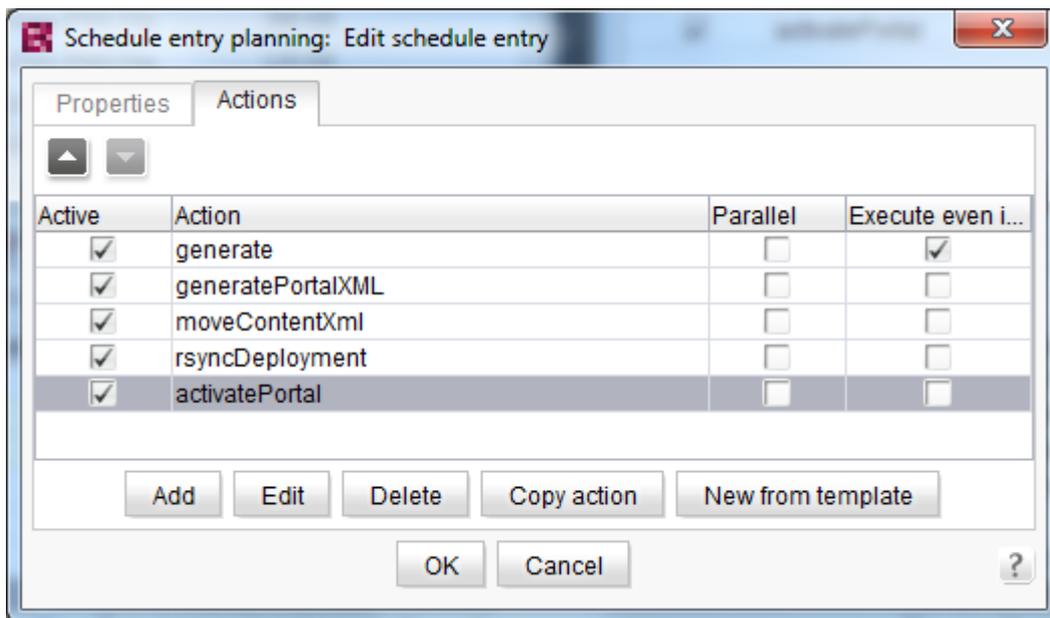


are addressed under the FriendlyURL.

**UID identifier prefix:** No function within Liferay

### 2.2.3 Configure schedule

The actions for generating, deploying and activating the information necessary for the portal can be created in the project's schedule management; here a specific order must be adhered to, which is shown in the following diagram.



**Figure 2-4: Action templates for portal integration**

#### Generating the portal files ("GeneratePortalXml")

The portal files are generated using the "GeneratePortalXml" script, which was created as an action template while the module was installed and can be created using the "New from Template" button.

Unlike references, copies can still be changed later, which is important in this case, as the portal entry points have to be added to the script later.

#### Move content.xml ("moveContentXml")

For proper, faultless display of all pages in the Liferay portal, it is necessary to move the content.xml generated in the first step a level higher in the root folder of the generated project. To do this, the script supplied under section 8.1 must be added to a schedule to be created manually using the "Run script" point in the dialog that



opens following "Add". The topic of content projections is described later in [Chapter 3.1.3](#).

### Generating the content ("Generate") / Deploying on the portal server ("Deploy")

These two actions correspond to the customary FirstSpirit generation or deployment, which can be added using the "Perform Generation" / "Perform Deployment" point in the dialog that opens following "Add".

The relevant deployment path must be given in the dialog window of the deploy schedule.

The deployment target to be entered must be a web application within the Liferay portal (Tomcat). The deployment can be carried out using the methods given in the projects (FTP, RSYNC, CRC Servlet).



*The generation folder must be located within the webapps folder of the Liferay Tomcat.*

### Activating the portal files ("activatePortal")

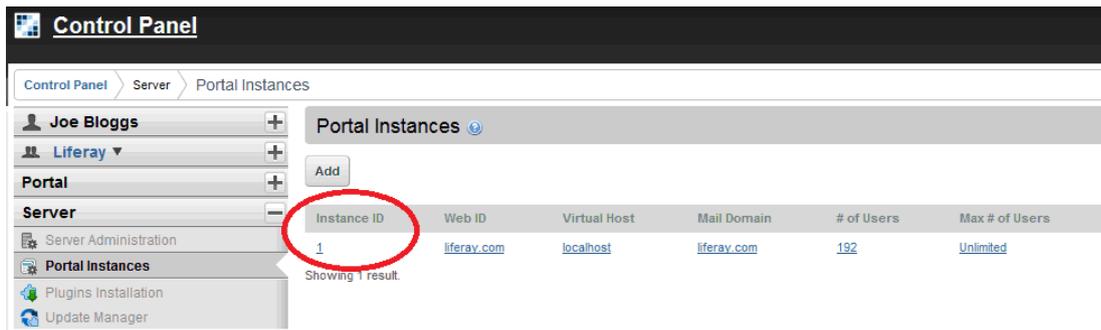
The portal pages are activated using the "ActivatePortalXml" script created as an action template on installation of the module; it can be created as a copy using the "New from Template" button. In this action, the script supplied under section 8.2 must be added, in which the following parameters must be adjusted to the user's own project.

```
javaPath = "C:/Program Files/Java/jdk1.6.0_18/bin/java";
liferayClient="C:/Program Files/FirstSpirit4/lifespirt.jar";
companyId="1234";
screen="test";
username="test@liferay.com";
password="test";
webserviceUrl="localhost:8080/LifeSpirit-6.0.5.1/secure/axis/";
deploymentDestination="C:/liferay/tomcat/webapps/mithras";
communityName= "Mithras - Liferay";
```



## Parameters to be adjusted

- **javaPath** Path to the Java JDK on the FirstSpirit Server required under 2.1.1
- **liferayClient** Here you enter the path for the Liferay web service client.
- **companyId** The portal instance ID given in the control panel of the Liferay portal

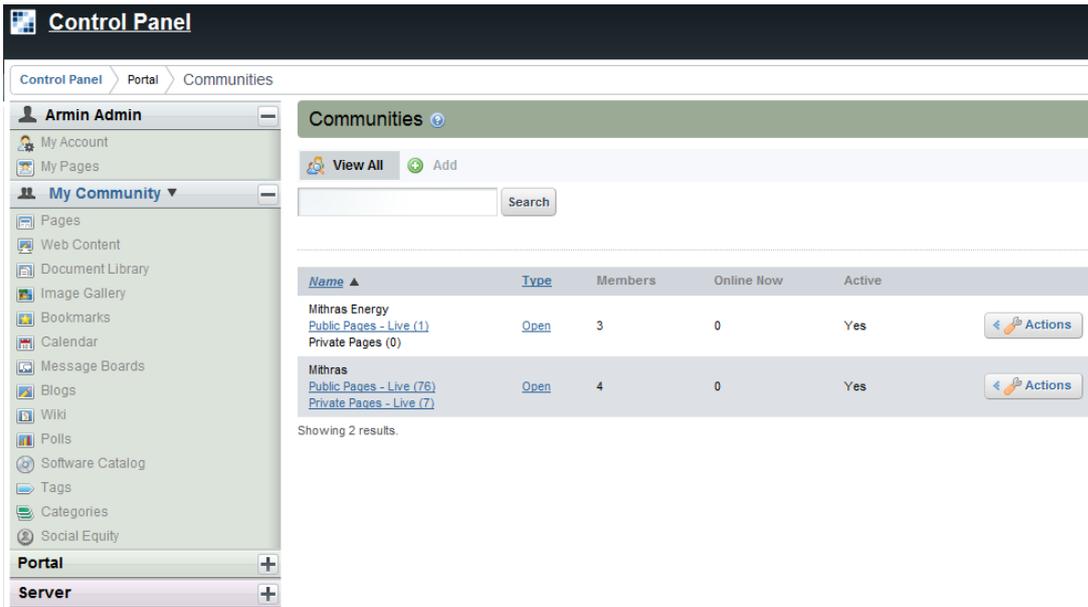


The screenshot shows the Liferay Control Panel interface. The breadcrumb navigation is 'Control Panel > Server > Portal Instances'. The user 'Joe Bloggs' is logged in. The left sidebar contains navigation options: Server Administration, Portal Instances (selected), Plugins Installation, and Update Manager. The main content area displays a table of Portal Instances. The table has columns: Instance ID, Web ID, Virtual Host, Mail Domain, # of Users, and Max # of Users. One instance is listed with Instance ID '1', Web ID 'liferay.com', Virtual Host 'localhost', Mail Domain 'liferay.com', # of Users '192', and Max # of Users 'Unlimited'. The value '1' in the Instance ID column is circled in red. Below the table, it says 'Showing 1 result.'

Instance ID	Web ID	Virtual Host	Mail Domain	# of Users	Max # of Users
1	<a href="#">liferay.com</a>	localhost	<a href="#">liferay.com</a>	192	Unlimited

- **screen** Here you must select the display name within the Liferay portal, in our example: test@liferay.com
- **username** The user name of the Liferay account set up under section 2.1.3
- **password** The password of the Liferay account set up under section 2.1.3
- **webserviceUrl**
- **deploymentDestination** The path given for the deploy schedule
- **communityName** Name of the community created in the Liferay portal



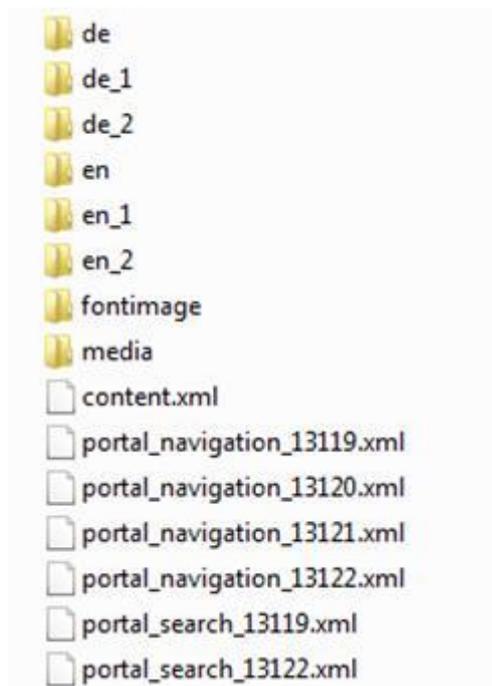


## 2.2.4 Deployment

After the pages have been created as required, they must be deployed in FirstSpirit in the usual way. As a result, the deployment schedules created under section 2.2.3 are executed one after the other.

If the deployment including all schedules is successful, all content folders, the XML files created for the entry points and the moved content file must be located under the path <Liferay root folder>/tomcat/webapps. For information about the topic of EntryPoints and their definition, please refer to the FirstSpirit Portal SAP EP BP 4.2 Documentation.





## 2.3 Adjusting personalization

In order to implement the topic of "Personalization" in the required Liferay project, various steps are necessary, which are explained in detail in the following.

### 2.3.1 Creating group including roles

Under the Configuration menu item in Server Monitoring, select the *PermissionService* under the existing services, which opens a list of various files.



The screenshot shows the FirstSpirit administration interface. At the top, there is a navigation bar with 'Language' dropdown, 'User Admin', and 'End monitoring'. A status bar shows 'FirstSpirit 5.0\_DEV.16.49732' and system metrics: Projects: 5, Sessions: 4, Actions: 0, Preview: 0, Memory: [ ], Load: [ ].

The main content area is titled 'Services' and contains a table with the following data:

Name	Comment	Type	Autostart
<a href="#">UXBService</a>	Uxb Service implementation.	UX Bridge	<a href="#">deactivate</a>
<a href="#">PackageManager</a>	PackagePool Service	System	<a href="#">deactivate</a>
<a href="#">PermissionService</a>	Permission Service	System	<a href="#">deactivate</a>
<a href="#">SpellService</a>	Spell Service	SpellService	<a href="#">deactivate</a>
<a href="#">TimeLineService</a>	TimeLine Service	System	<a href="#">activate</a>
<a href="#">TestService</a>	Test Service	System	<a href="#">activate</a>
<a href="#">FaceAppService</a>		FaceApp	<a href="#">deactivate</a>
<a href="#">EnterpriseSearchService</a>		Exalead-Push	<a href="#">deactivate</a>
<a href="#">Comments Service</a>	Service component for getting data via webservice technology.	SocialWeb	<a href="#">deactivate</a>
<a href="#">OfficeImportService</a>	Office Service Configuration serving the definitions for server global import rule sets.	FirstSpirit Office	<a href="#">deactivate</a>
<a href="#">ConfigTestService</a>	Configuration Test Service	FIRSTpersonalisation	<a href="#">deactivate</a>
<a href="#">FOP-Font-Info</a>	FOP Font Info Service	Apache FOP v0_20_5	<a href="#">deactivate</a>

A group is created either by creating a new XML file or by changing *groups.xml*.

The corresponding file must contain the following code.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<GROUPS name="Example" version="1">
<GROUP id="User" name="User"/>
<GROUP id="PowerUser" name="Power User"/>
</GROUPS>
```

The first parameter *name* represents the name of the group, all other the names of the roles belonging to the group.

*Please note that the IDs of the roles must correspond to the role names of the Liferay portal!*



**Roles** ⓘ

View All + Add

Search

---

Name ▲	Type	Description	Actions
Administrator	Regular	Administrators are super users who can do anything.	⏪ Actions
Community Administrator	Community	Community Administrators are super users of their community but cannot make other users into Community Administrators.	⏪ Actions
Power User	Regular	Power Users have their own public and private pages.	⏪ Actions
User	Regular	Authenticated users should be assigned this role.	⏪ Actions

It is also possible to use the "Add" button to create new roles in Liferay, which can be assigned in FirstSpirit. For these, select the type "Regular".

**Roles** ⓘ

View All + Add

**New Role**

Name  
TestUser

Title  
Other Languages (0) ▼

Description

Type  
Regular ▼

Save Cancel

### 2.3.2 Entering group

To use the new group created, it is necessary to extend the service.ini, which can also be found under the "PermissionService" point in Server Monitoring, to include the corresponding group entries.



```
#
# global params
# -----

# check each x seconds for changes
interval=20

# symbolic names for documents
documents=GroupsFile,Example

#
# document specific params
# -----
GroupsFile.path=groups.xml
Example.path=example.xml
```

On the one hand, the name of the new group created must be passed as *document* and on the other the path to the XML file must be given.

### 2.3.3 Using group

If the service.ini has been extended accordingly, the group can then be used in FirstSpirit.

To do this, the `$CMS_INPUT_PERMISSION$` component – preferably in the template of the metadata – must be used, in which the name of the new group created is to be passed as *group*.

```
<CMS_INPUT_PERMISSION name="perms" group="Example" hFill="yes">
  <LANGINFOS>
    <LANGINFO lang="*" label="Permissions" />
  </LANGINFOS>
</CMS_INPUT_PERMISSION>
```

Then, different roles can be defined for the roles of the group in the Site Store and in the Media Store.





**All permissions must initially be set on the root nodes of the Site Store and Media Store, to avoid generation errors.**



If the roles have been assigned to a page belonging to an EntryPoint, they are also listed in the XML file of the EntryPoint created by the deployment.

```
<Item name="startpage" version="16439"  
modifiedby="Admin" pageversion="16434"  
roles="'User', 'PowerUser'"  
type="mixedportalpage">
```



### 3 Extending the FirstSpirit project

To link the Liferay portal to FirstSpirit various templates are required, which have only partly already been added by the installation of the module and all of which contain the following form element.

#### 3.1 Page templates

For proper, faultless display of the different pages in the portal, two different page formats exist, which are listed here in detail.

In addition to the form element already described in the previous section, both page templates each also require a radiobutton, which only differs in value, on the basis of which the page type is transferred.

##### Possible value

- **portalpage** Value for a portal page
- **mixedportalpage:** Value for a mixed portal page

```
<CMS_INPUT_RADIOBUTTON
  name="fs_portal_pageType" gridHeight="1" gridWidth="1"
  hFill="yes" hidden="no" noBreak="no" useLanguages="no">
  <ENTRIES>
    <ENTRY value="_PLEASE_SELECT_THE_TYPE_OF_PAGE_">
      <LANGINFOS>
        <LANGINFO lang="*" label="Application"/>
      </LANGINFOS>
    </ENTRY>
  </ENTRIES>
  <LANGINFOS>
    <LANGINFO lang="*" label="Portal"/>
  </LANGINFOS>
</CMS_INPUT_RADIOBUTTON>
```

Further, the following code must be added in the HTML tab; this code causes the content of the relevant page to be transcribed in an XML text that can be



implemented by Liferay. Here it is necessary to note that each individual section of the page must be output in a section tag.

```
<firstspirit>
<metainfo>
  <description><![CDATA[Here you find our
contacts]]></description>
  <keywords><![CDATA[Contacts]]></keywords>
  <robots><![CDATA[index, follow]]></robots>
  <htmltitle><![CDATA[Contacts TSS Austria]]></htmltitle>
  <friendlyurl>
    <![CDATA[$CMS_REF(ref(#global.ref).url)$]]>
  </friendlyurl>
</metainfo>

$CMS_FOR(section,
#global.page.body("left").getChildren(class("de.espirit.firstspiri
t.access.store.pagestore.Section")))$
<section id="$CMS_VALUE(section.getId())$"><![CDATA[
$CMS_VALUE(section)$
]]></section>
$CMS_END_FOR$

$CMS_FOR(section,
#global.page.body("content").getChildren(class("de.espirit.firstsp
irit.access.store.pagestore.Section")))$
<section id="$CMS_VALUE(section.getId())$"><![CDATA[
$CMS_VALUE(section)$
]]></section>
$CMS_END_FOR$

</firstspirit>
```

The following metadata parameters, which correspond to the HTML metatags of a page, must be adjusted:

- **description** Short description of the page
- **keywords** Topics or key words dealt with on the page



- **robots** Instruction for search robots to include the page in their index
- **htmltitle** Title of the page
- **friendlyurl:** The pages can be reached within Liferay under the following URL, therefore, the pages should be filled with the respective FirstSpirit URL. If the respective URL differs from the URL generated by FirstSpirit in order to generate an SEO optimized URL, accordingly, the links on the following page must be generated correctly.

### 3.1.1 Portal page

Unlike the other page format, in this format, only pure FirstSpirit sections may be used. On deployment, these are grouped together to form a single large block and can therefore no longer be individually edited in the portal.

The screenshot shows a Liferay portal page for Mithras Energy. The page has a navigation bar with 'Company', 'Departments', and 'Services'. The main content area is divided into several sections:

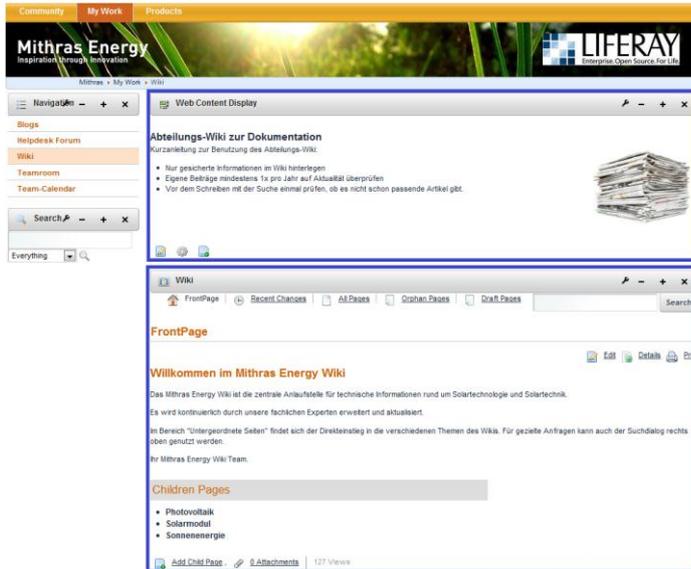
- Company philosophy:** A quote: "When the wind of change blows, some build walls and the others build windmills." We believe in changes and positively look forward to them. Only those who make the developments in the world their own can achieve good things with them. This is also how we observe innovations in the solar industry and effectively implement them in our products. In this way, our customers can always benefit from our applications. Future means change. Change which we can turn into something positive. We face this challenge. Always. In order to develop the best solutions for you. (Accompanied by an image of a modern building.)
- Objectives:** With our products, we want to significantly raise the quality of products offered on the solar market. Our standards are already used as a European guideline. We want to achieve this status for the rest of the solar energy industry too and therefore set trends. Solar products are only really efficient if they are preceded by intelligent developments and satisfy the highest quality standards. We consistently support this principle and will implement it as the standard for other countries too. To realise this, we conscientiously work on our developments and the communication measures which publish this for the non-European region.
- News:** A list of recent news items:
  - 27-Jan-2011: Mithras Energy receives solar prize from the City of Sonningen
  - 10-Jan-2011: New Director of Mithras Energy
- FirstSpirit:** The mithras-energy.de website is a demo project of e-Spirit AG, Dortmund. We have compiled more information on the FirstSpirit content management system used for you.

### 3.1.2 Mixed portal page

With this format is possible to use both FirstSpirit and Liferay elements together on a page. These are all considered as individual portlets in the portal and are **not** grouped together to form a block, as on the portal page. Individual editing is therefore also possible in the portal.

This format is therefore particularly important for the personalization of individual sections.





### 3.1.3 Content projection and other loose pages

To deploy all loose pages, which cannot be reached within the navigation (start pages), which are not already covered by a navigation.xml or search.xml, an XML file is required, which is created by this page template. It must have the name *content*, be filed in the Site Store in a folder with the reference name "system" and generate an XML file with the following structure.

```
<Root id="#global.project.id" name="#global.project.name">
  <Item name="xyz" version="12345" pageversion="9876"
    parent="parent"
    roles="'Guest','Customer','Marketing','Member'">

    <Title locale="de">title_de</Title>
    <Url locale="de">
      /URL/
    </Url>

    <Title locale="en">title_en</Title>
    <Url locale="en">
      URL
    </Url>
  </Item>
</Root>
```



**name:** Identifier / reference name

**version:** Revision of the page reference

**pageversion:** Revision of the page

**parent:** Parent node in which the page is hung

**roles:** List of all roles that can access the page.

**title:** Title of the navigation

**Url:** Url of the generated page

### 3.1.4 Project settings

The project settings must be extended to include a text input field with the name "fs-liferay-communityname", in which the editor must give the name of the community crated in the portal. This is required for subsequent link generation.

```
<CMS_INPUT_TEXT name="ps_liferay_community_name" hFill="yes"
    singleLine="no" useLanguages="no">
    <LANGINFOS>
        <LANGINFO lang="*" label="Community name" />
    </LANGINFOS>
</CMS_INPUT_TEXT>
```



The screenshot shows a form field with a light gray header containing the text "Community name". Below the header is a text input field with a thin border and a dashed focus outline. The text "Mithras Energy" is entered into the field.



## 3.2 Portlet sections

The section template Portal\_Portlet provides the option for integrating Liferay portlets into a page.

### 3.2.1 Determining portlet type

A portlet is positioned by a selection, which provides the IDs of the different Liferay portlets on the basis of the individual values. The form element of this selection must be named with *"fs\_portal\_targetId"*.

#### Page type

If a section is a portlet, this must be given by a corresponding form component, which must be named *"fs\_portal\_pageType"*. Preferably, a selection with only one value is offered here (radiobutton or similar). After it has been created the relevant entry must be defined by default value and should then be set to hidden, to avoid editorial change.

```
<CMS_INPUT_RADIOBUTTON
  name="fs_portal_pageType"
  gridHeight="1" gridWidth="1" hFill="yes"
  hidden="yes" noBreak="no" useLanguages="no">
  <ENTRIES>
    <ENTRY value="portlet">
      <LANGINFOS>
        <LANGINFO lang="*" label="Portlet"/>
      </LANGINFOS>
    </ENTRY>
  </ENTRIES>
  <LANGINFOS>
    <LANGINFO lang="*" label="Portal"/>
  </LANGINFOS>
```





Examples of possible IDs:

Search: 3 / Calendar: 8 / Forum: 19 / Bookmarks: 28 / Blog: 33 / Wiki: 36 /  
Login: 58

```

<CMS_INPUT_COMBOBOX
  name="fs_portal_targetId" convertEntities="standard"
  editable="no" hFill="no" noBreak="no" useLanguages="no">
  <ENTRIES>
    <ENTRY value="3">
      <LANGINFOS>
        <LANGINFO lang="*" label="Search"/>
      </LANGINFOS>
    </ENTRY>
    <ENTRY value="19">
      <LANGINFOS>
        <LANGINFO lang="*" label="Forum"/>
      </LANGINFOS>
    </ENTRY>
  </ENTRIES>
  <LANGINFOS>
    <LANGINFO lang="*" label="Portlet/iView-ID"
      description="Please select the desired
      Portlet/iView."/>
  </LANGINFOS>
</CMS_INPUT_COMBOBOX>
</CMS_INPUT_RADIOBUTTON>

```

### 3.3 Linking

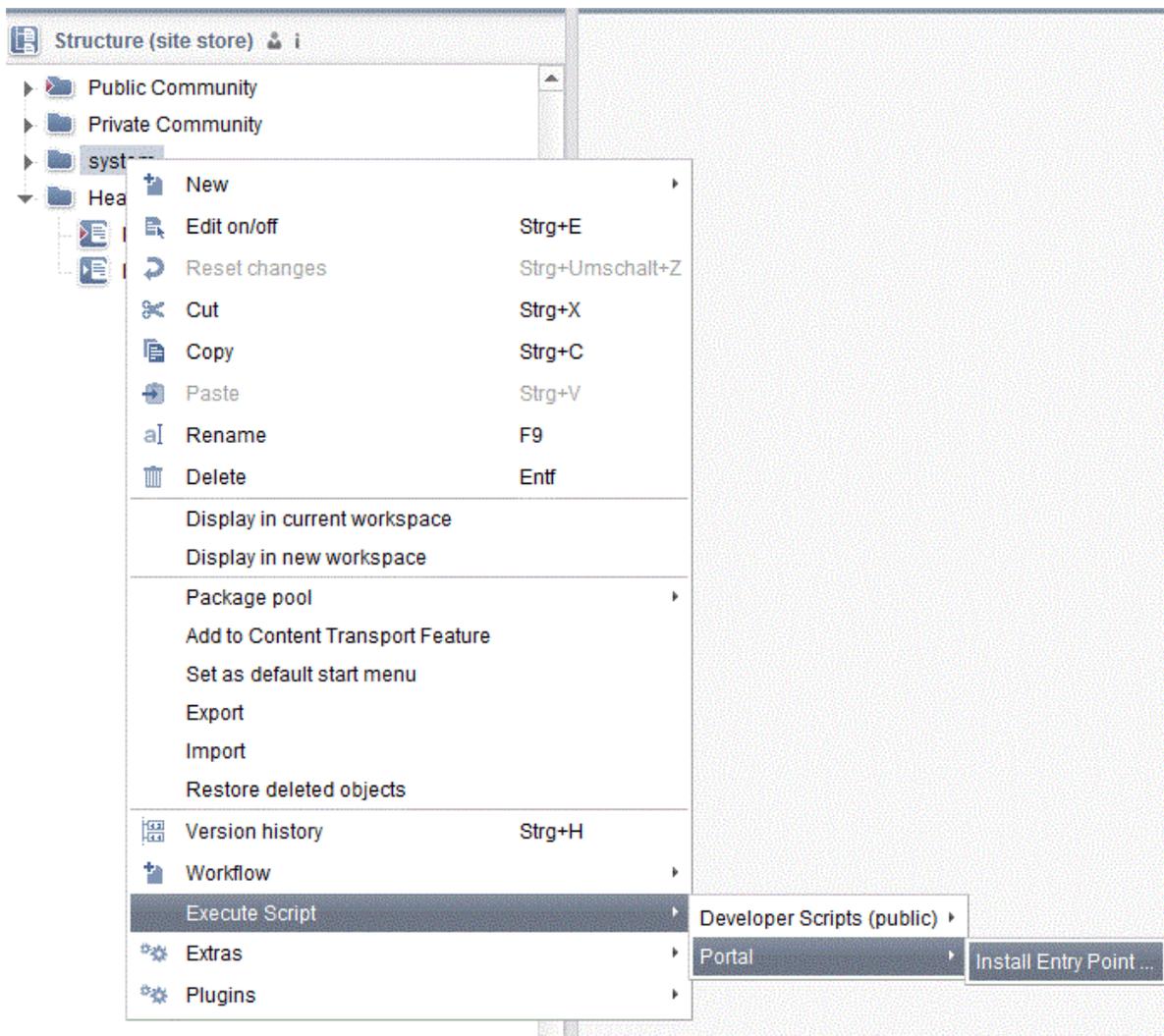
By defining the FriendlyURL within the metablock of a page template, the URL within Liferay can be determined. SEO requirements can also be implemented in this way. However, the URL generated within the pages via link templates, must match the FriendlyURLs set within the pages, as otherwise it is not guaranteed that the pages can be reached. If the URL differs from the entry given by FirstSpirit, the relevant link templates must be adjusted in the same pattern, to ensure link consistency. If it is possible to change between a "private" and "public" community within the project,



this must be implemented within the URL by the prefix "/web/" or "/group/". In this case, the respective link templates can read out the project setting or the Site Store variable.

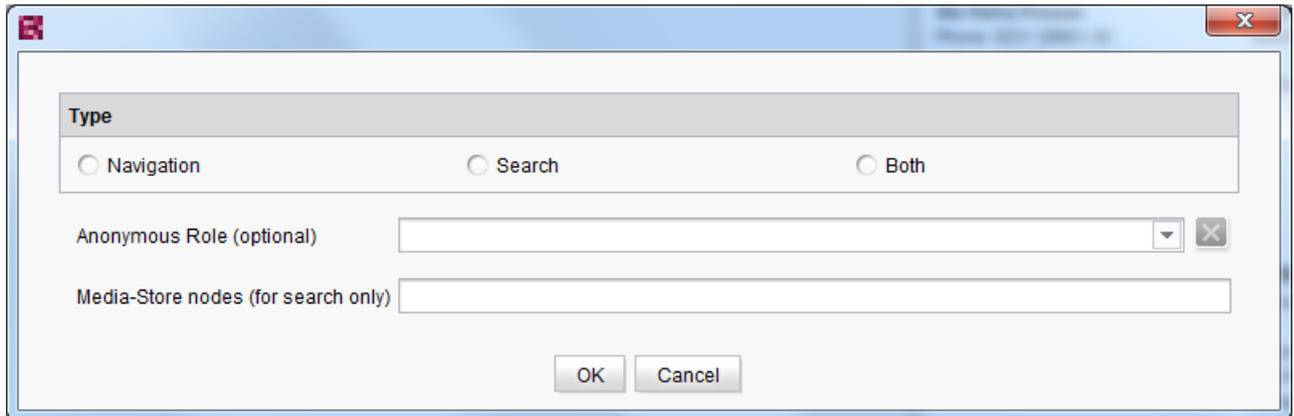
### 3.4 Defining EntryPoints

To transfer navigation into the Liferay portal, the individual navigation items must be defined as EntryPoints in the Site Store of the FirstSpirit project. This takes place via a script, which was automatically created when the "FirstSpirit Portal" module was installed and which can be run with a right-click.



In the dialog that then opens, it is necessary to specify whether the EntryPoint only relates to the navigation or search or to both.





The dialog box has a title bar with a close button. Below the title bar is a section labeled 'Type' containing three radio buttons: 'Navigation', 'Search', and 'Both'. Below this are two input fields: 'Anonymous Role (optional)' with a dropdown arrow and a close button, and 'Media-Store nodes (for search only)'. At the bottom are 'OK' and 'Cancel' buttons.

If there are several pages in the selected folder, only the start page is taken into account in the corresponding *navigation.xml*, while all pages in the folder are covered in the corresponding *search.xml*.

After installing the EntryPoint, various structure variables are created for the selected folder, which are filled depending on the input.

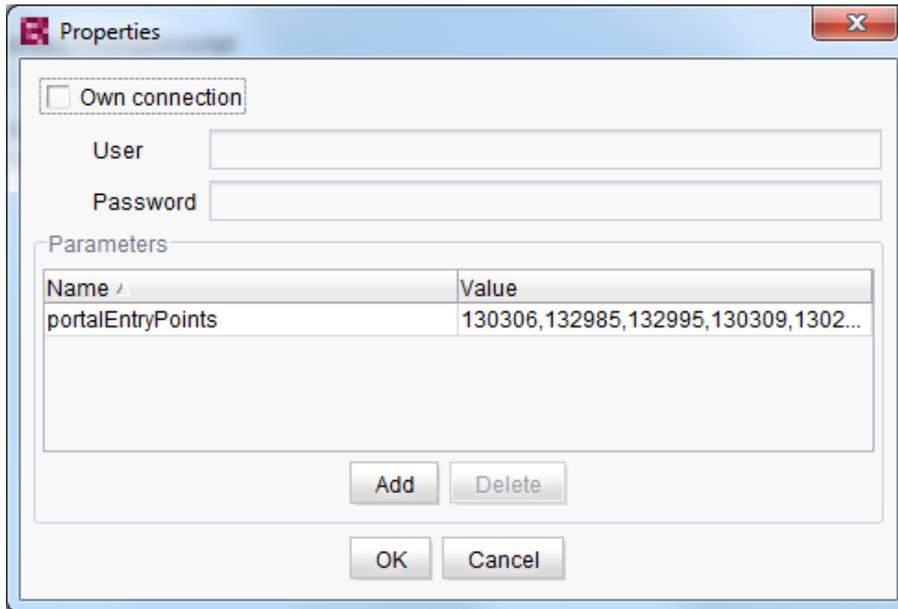
Name	Wert
fs_portal_anonymousRole	PowerUser
fs_portal_entryPoint	true
fs_portal_mediaNodes	
fs_portal_xmlFactory	search

The FirstSpirit ID of the selected navigation item must then be manually added to the "GeneratePortalXml" schedule in the project configuration.



*If several EntryPoints have been defined, the IDs must be transferred in a comma-separated list (**without** spaces).*





Name	Value
portalEntryPoints	130306,132985,132995,130309,1302...

The script creates different XML files in the corresponding generation folder, which are required by Liferay to create the navigation.

### 3.5 Handling media in the Liferay portal

If file extensions of media were given for the *Media extensions* entry in the configuration made under section 2.2.2 and an *EntryPoint* of the type *Search* was set on a structure folder containing media of this type, all media referenced by this are imported into the Liferay portal. This requires the setting of permissions on the root folder of the Media Store. Therefore, a corresponding media can also be found by a search through its content. Further, it is possible to display these media only under consideration of the permissions defined in the personalization.

If an *EntryPoint* of the type *Search* has been set on a structure folder containing a medium, but no corresponding file extensions were given in the configuration described under section 2.2.2, the referenced medium can only be found by its heading with the search in the Liferay portal, as in this case only a link to the medium is generated, without importing the medium itself into the portal. Instead, it is filed in the project's generation folder.





*As the permissions set on the root folder of the Media Store are taken into account when media are imported, it does not make any sense to give the file extensions of images in the configuration described under section 2.2.2. These would then also only be displayed depending on the permissions set, which would result in unwanted behavior.*

## 3.6 Metadata

These attributes are used to fill the metadata which, for example, can be used for the search. Here it must be noted that there are two options for transferring attributes.

### Option 1: Transfer precisely one attribute

If precisely one attribute only is to be transferred, the corresponding form component must begin with the name "fs\_portal\_attribute\_", to which a freely selectable term is to be appended.

```
<CMS_INPUT_TEXT
  name="fs_portal_attribute_abc" convertEntities="standard"
  hFill="yes" length="38" maxLength="100" noBreak="no"
  useLanguages="yes">
  <LANGINFOS>
    <LANGINFO lang="*" label="Parameter Abc" description="Please
enter the parameter 'Abc'."/>
  </LANGINFOS>
</CMS_INPUT_TEXT>
```

### Option 2: Transfer a list of attributes

If several attributes are to be transferred, the corresponding form element must be named with "fs\_portal\_attributes".

```
<CMS_INPUT_TEXTAREA
  name="fs_portal_attributes" columns="20" rows="5"
  convertEntities="standard" hFill="yes" noBreak="no"
  useLanguages="yes">
  <LANGINFOS>
    <LANGINFO lang="*" label="Please enter the parameters of the
Portlets/iViews." description="."/>
  </LANGINFOS>
```



```
</CMS_INPUT_TEXTAREA>
```



*At present, the attributes are not processed by Liferay. Therefore, the entry of attributes is only an anticipatory measure.*



## 4 Differentiation between private & public community

In its portal, Liferay differentiates between a public and a private community. While only logged-in persons have access to the private area, the public area is visible to every user – including guests.

This differentiation is established in FirstSpirit by assigning two different values to a new structure variable to be created. Here it must be noted that the *web* value predetermined by Liferay corresponds to the public community and the *group* value corresponds to the private community.

liferay_area	web
liferay_area	group

The obvious thing to do is to divide the structure itself accordingly in the FirstSpirit project in order to be able to set the navigation points accordingly.

These variables must then be read out by the link template, in order to generate the respective correct FriendlyURL.



*Additional, optional rules for the visibility can be defined within the portal.*



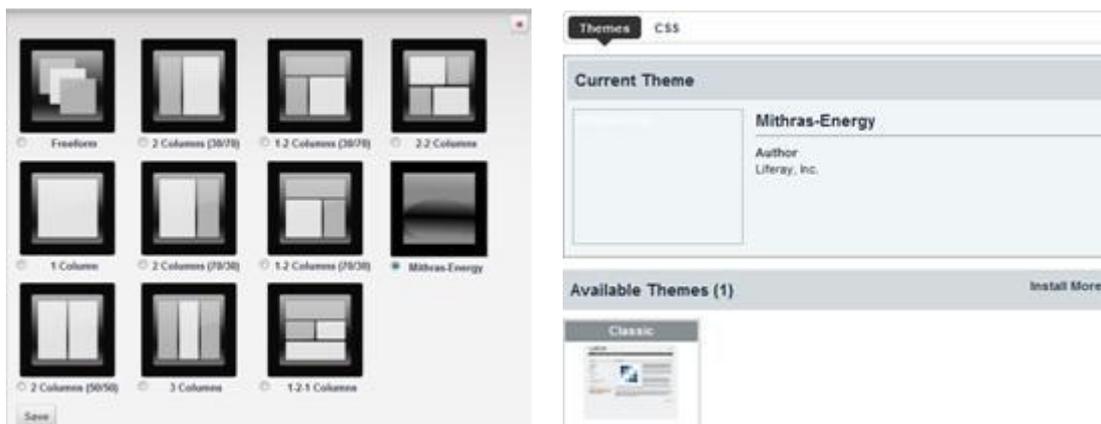
## 5 Layout & theme

It is possible to choose between different layouts and themes for individual design of a project in the Liferay portal. The layout defines the arrangement of the individual page content in a variable number of columns named containers, while the theme is responsible for their display.

By default, Liferay provides several layouts and themes of its own, which you can choose in any way you wish. However, it is also possible to use your own layouts and themes.



For information on installing and selecting layouts or themes you have created yourself, please refer to [the Installation Guide](#).



The Mithras Layout used here by way of example provides a choice between two and three-column display. It must also be possible to make this selection in FirstSpirit (e.g. via a ComboBox). To do this, the page templates must be extended to include the following code.

```
<CMS_INPUT_COMBOBOX
  name="fs_portal_layoutTemplate" convertEntities="standard"
  editable="no" hFill="yes" noBreak="no" useLanguages="no">
<ENTRIES>
  <ENTRY value="Mithras_Energy">
    <LANGINFOS>
      <LANGINFO lang="*" label="2 column Mithras"/>
    </LANGINFOS>
  </ENTRY>
</ENTRIES>
```



```
</ENTRY>
<ENTRY value="3_columns">
  <LANGINFOS>
    <LANGINFO lang="*" label="3 column Mithras"/>
  </LANGINFOS>
</ENTRY>
</ENTRIES>
<LANGINFOS>
  <LANGINFO lang="*" label="Layout"/>
</LANGINFOS>
</CMS_INPUT_COMBOBOX>
```

The corresponding form component must be named *fs\_portal\_layoutTemplate* and should be set to hidden if the layout for this page type is selected and fixed via the default values and it should not be able to be changed by the editor.

Further, the content areas available in FirstSpirit must be assigned to the corresponding column of the layout in the Liferay portal. This must be done using an input component named *fs\_portal\_layoutContainer* which, depending on the previous decision regarding the *fs\_portal\_layoutTemplate* component, should also be hidden from the editor following entry of the default values.

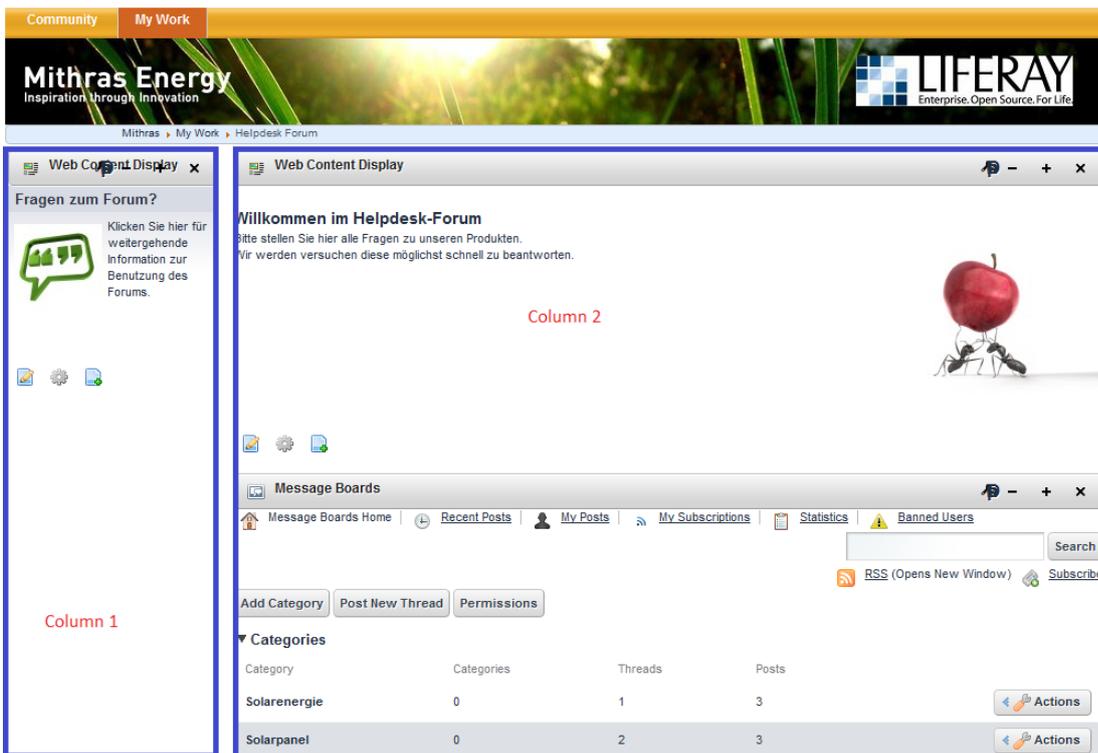
```
<CMS_INPUT_TEXTAREA
  name="fs_portal_layoutContainer"
  columns="20" convertEntities="standard" hFill="yes"
  hidden="yes" noBreak="no" rows="5" useLanguages="no">
  <LANGINFOS>
    <LANGINFO lang="*" label="Container"/>
  </LANGINFOS>
</CMS_INPUT_TEXTAREA>
```



E.g.: Assignment of the FirstSpirit content area "Content" to the Liferay layout column "column 2".



 Please note that the first column of the layout is basically reserved for the navigation.



Workspace Company Departments Services

Mithras Energy  
Inspiration through Innovation

LIFERAY  
Enterprise. Open Source. For Life.

Mithras > Workspace

NAVIGATION

Web Content Display x

Contact

Please contact us for more information:

Ms Petra Presse  
Phone: 0231 28661-30  
e-Mail: info@e-spirit.de

Search - + x

Everything

Column 1

Web Content Display - + x

Message from the Board

New Director Hans Energie of Mithras Energy.

Since 1 January 2012, the executive management of Mithras Energy Solartechnik GmbH has been reinforced by Hans Energie. This is the company's response to the growth of recent months and reflects the success of the company.

More Information

Web Content Display - + x

Top News

Mithras Energy receives solar prize from the City of Sonningen.

Mithras Energy was awarded the Sonnigen Solar Prize by Mayor Karl-Heinz Schmidt during an official ceremony. The prize honours the company and its constant commitment to solar energy in German speaking countries. The managing director of Mithras Energy received this prize gratefully.

More Information

Column 2

Tag Cloud - + x

cebit deutschland firstspirit  
liferay photovoltaik  
solarmodul solarzelle  
sonnenenergie  
stromerzeugung  
termin

Web Content Display - + x

Helpdesk

Any questions about our products? Please visit our Helpdesk Forum.

Web Content Display - + x

Calendar

My Team Calendar

Column 3



## 6 Creating and subsequent deployment

This chapter explains the creation and subsequent transfer of a blog with preceding introductory text by way of an example.

### 6.1 Creating a page

As in usually the case in FirstSpirit, first of all, a page must be created in the Page Store. As this is to contain both an introductory text and a Liferay portlet, it must be of the type Mixed Portal Page.

The default values, not visible to the editor, define that the page is a mixed portal page, its layout and how the individual columns of the layout are named.



Preview : Portalseite (1 Spalte + Marginal)

Fall-back values DE EN

Extra Page

Portal

Mixed Portal Page

2 Spaltig Mithras

content=column-2  
left=column-1

Portalpage Properties

Use standard navigation portlet?  Yes  No

Allow Portlets in marginal column?  Yes  No

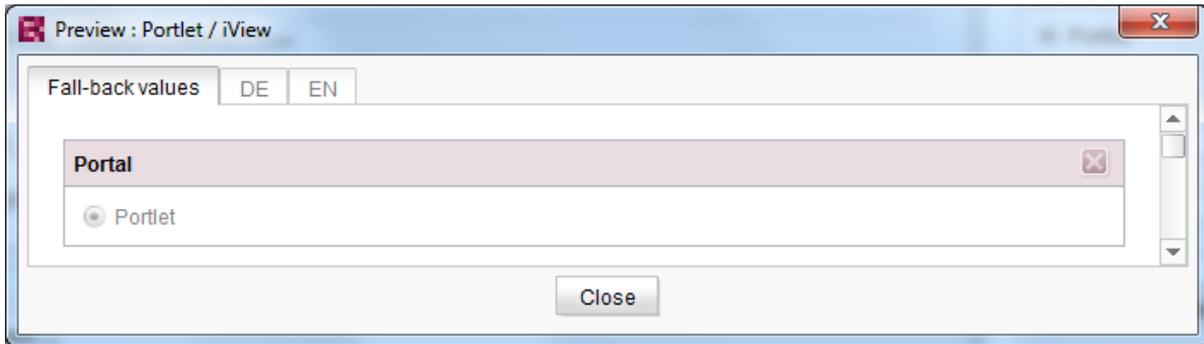
Create section

Close

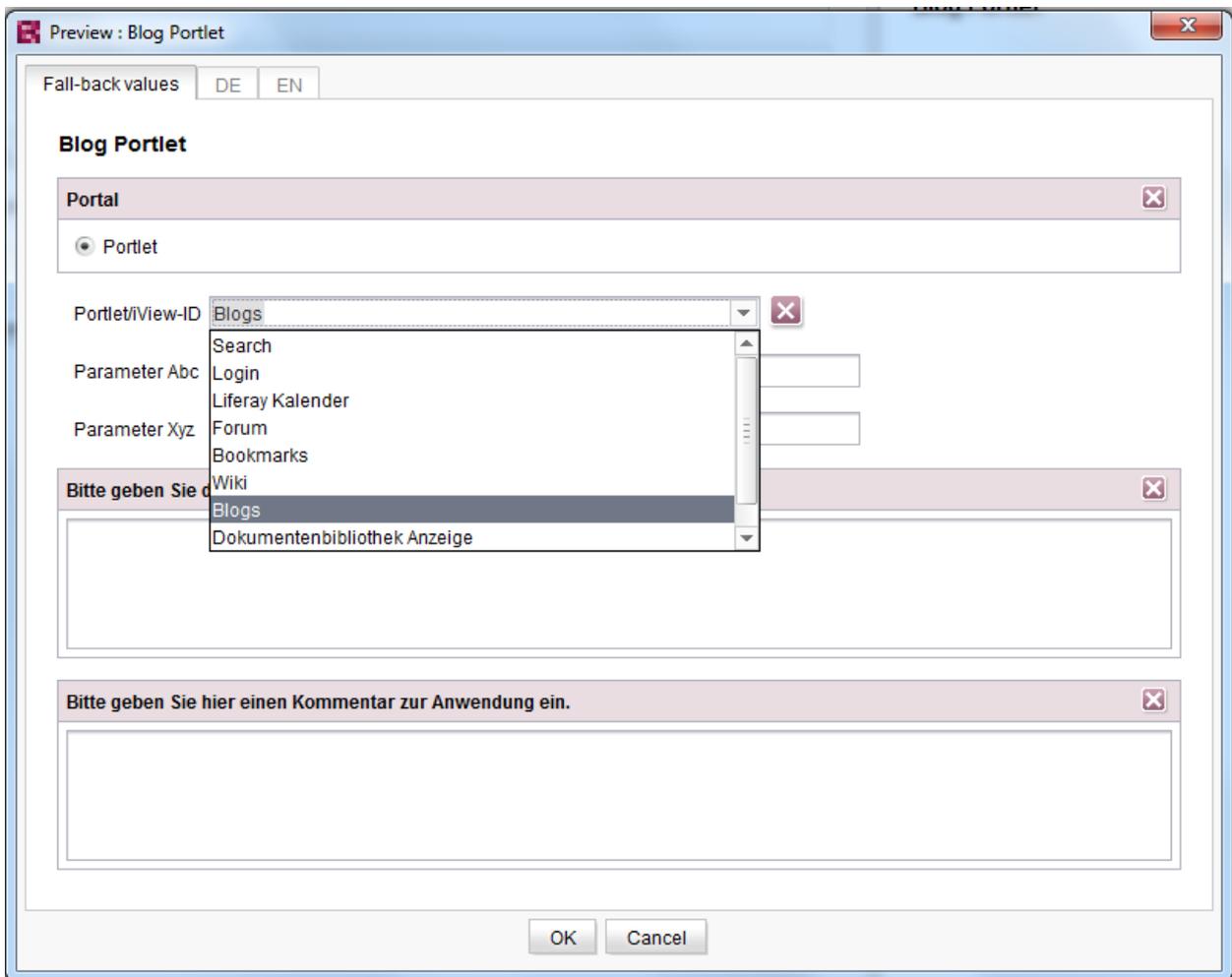
A text section containing the introductory text and a portlet/iView section are then added. If the page is to display other text and/or other portlets, they can be added as required. The position of the individual sections is, as usual, freely selectable.

The concealed default values of the portlet section merely define that it is a portlet.





The choice of required portlet is made by the editor. Parameter details and a comment are optional and are currently not processed by the Liferay portal. These are merely anticipatory measures.

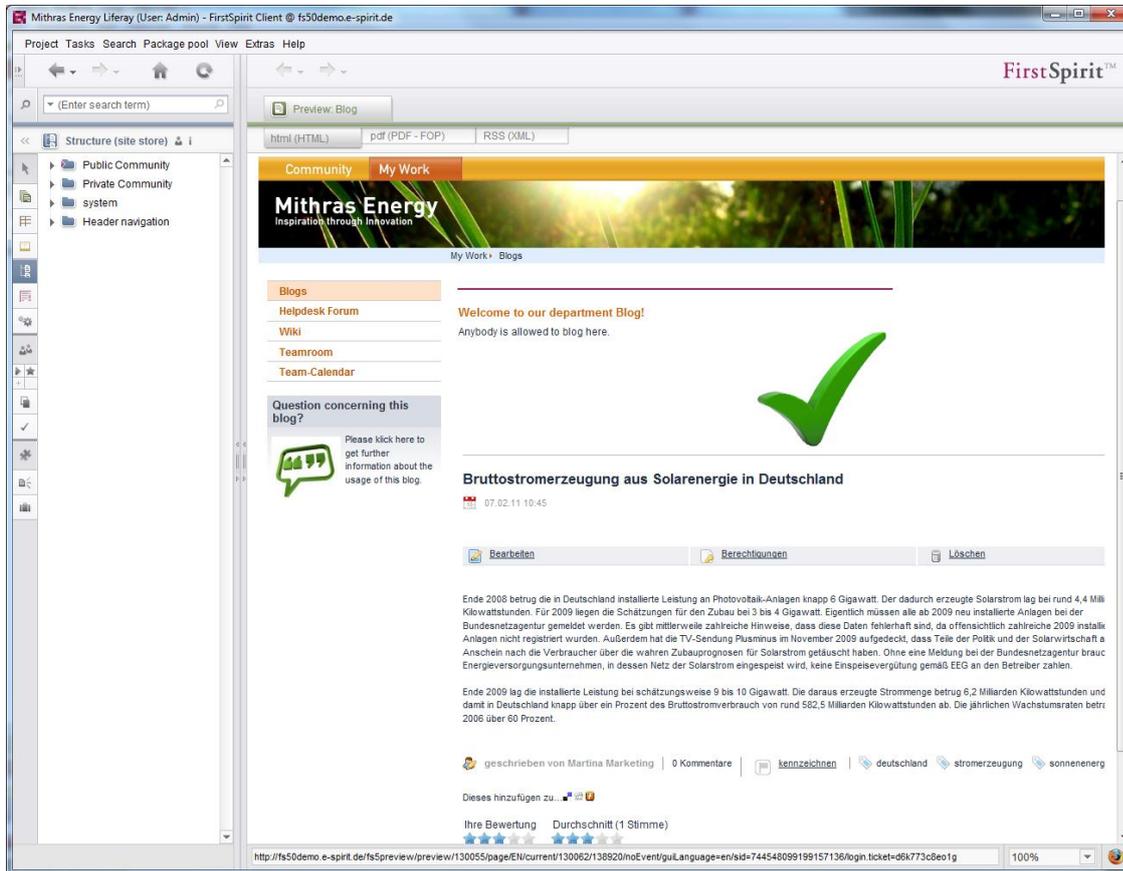


If the created page is to be accessible via the navigation in its deployed state, the structure folder that contains it must be defined as an entry point - as described under section 3.3.



## 6.2 Result in the portal

Following successful deployment the page should also be integrated in the relevant place in the portal and in the case of the exemplary blog page resembles the following display.



## 7 Legal notices

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

The user can only use the module as defined under the terms of the license agreed with e-Spirit AG.

Details of possible external software products used, not produced by e-Spirit AG, their own licenses and any update information, is given on the start page of each FirstSpirit server, in the "Legal Notices" area.



## 8 Scripts

### 8.1 moveContentXml

```
#!/Beanshell
import java.io.*;

src = context.getPath() + "de/system/content.xml";
dst = context.getPath() + "/content.xml";

temp = new File(dst);
source = new File(src);

context.logInfo("Source:" + source);
context.logInfo("Dest: " + temp);

if(!source.exists()) context.logError("Source file not exists!");

if(temp.exists()) {
context.logInfo("File exists!");
temp.delete();
}

if(!source.renameTo(temp))
context.logError("Can't rename file");
```

### 8.2 activePortal

```
#!/Beanshell
import java.util.*;

t = context.getTask();
params = t.getParameters();
javaPath = '"C:/Program Files (x86)/Java/jdk1.6.0_18/bin/java"';
liferayClient='"C:/Program Files
(x86)/FirstSpirit4/lifespirt.jar"';
```



```
companyId="1";
username="test";
password="test";
webserviceUrl="localhost:8080/LifeSpirit-6.0.5.1/secure/axis/";
deploymentDestination="C:/Users/pahl/Documents/liferay-portal-
6.0.6/tomcat-6.0.29/webapps/mithras";
communityName="Mithras - Liferay - Test";
privateFlag="false";
mode="7";
overwrite="false";

class StreamCapture extends Thread {
    InputStream i;
    boolean useErrlog;
    StreamCapture(InputStream inputstream, boolean useErrorlog) {
        i = inputstream;
        useErrlog = useErrorlog;
    }
    public void run() {
        try {
            InputStreamReader reader = new InputStreamReader(i);
            BufferedReader buffr = new BufferedReader(reader);
            String l = null;
            while ((l = buffr.readLine()) != null) {
                if(useErrlog) {
                    context.logError(log + l);
                } else {
                    context.logInfo(log + l);
                }
            }
        } catch (IOException ex) {
            context.logError(log + " failed: " + ex);
        }
    }
}

try {
    String[] cmd = {javaPath, "-jar", liferayClient, companyId,
username, password, webserviceUrl, deploymentDestination,
```



```
communityName, privateFlag, mode, overwrite};
    context.logInfo("Starte Prozess: " + cmd.toString());
    Process p = new ProcessBuilder(cmd).start();
    InputStream is = p.getInputStream();
    StreamCapture stdout = new StreamCapture(p.getInputStream(),
false);
    StreamCapture stderr = new StreamCapture(p.getErrorStream(),
true);
    stdout.start();
    result = p.waitFor();
    if ( result == 0 ) {
        context.logInfo("Liferay Deployment successfully updated.");
    } else {
        context.logError("Error while updating Liferay");
    }
} catch (Exception e) {
    context.logError("Error while updating Liferay: " + e);
}
```

