FirstSpirit™
Unlock Your Content

FirstSpirit™ EmailMarketing
FirstSpirit™ Version 4.x, 5.x

Version	**2.1**
State	**RELEASED**
Date	**2013-02-19**
Department	Product Management
Copyright	2012 e-Spirit AG
File name	NEWS4xEN_FirstSpirit_Modules_Newsletter

e-Spirit AG

Barcelonaweg 14
44269 Dortmund | Germany

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

info@e-spirit.de
www.e-spirit.de

e-Spirit

Table of contents

1 Introduction

The *FirstSpirit™ EmailMarketing* module described in this document provides a convenient function for creating newsletters and other mailings in FirstSpirit™. Through the seamless interfacing of the "Universal Messenger" software of the firm "Pinuts media + science", it is possible to start mailing campaigns directly from FirstSpirit™. This document describes the installation and configuration of the *EmailMarketing Module*, and the installation as well as the necessary settings of the Universal Messenger. Equally, it describes how to send a newsletter and a test newsletter from FirstSpirit™.

A FirstSpirit™ project exists in addition to this documentation, in which the described functions are implemented by way of example. In addition, the project contains examples of section templates for newsletter registration, user profile management and password recovery. The example project can be imported via the FirstSpirit™ server and project configuration.

> ! *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 Functional scope

The add-on described here makes the following functions available to the editor in FirstSpirit™:

- Creation and maintenance of newsletters & other mailings
- Sending of mailings
- Test dispatch of mailings

The module's functions do not include the management of subscriber data, analyses of statistics or bounce management, as these functions are provided by the web interface of the Universal Messenger, which can be opened from the FirstSpirit™ AppCenter.

1.2 Function

The e-mail marketing software Universal Messenger (UM) is used to send mailings via FirstSpirit™. With the help of UM, distribution lists of so-called channels can be created and subscribers can be managed. The UM can be used directly as a web application, or its jobs (schedules) are accepted from an external application. The module described here utilises this function of UM and enables FirstSpirit to use UM to send mailings.

In order to process an external job, UM requires first, the content of the mailing to be send and secondly, an event file, which contains all the necessary information for sending the newsletter. This event file is then copied into a folder monitored by UM. UM uses the event file to work through the schedule and starts to send the mailings immediately or with time-controlled dispatch.

1.3 How it works in the FirstSpirit context

The FirstSpirit project must be extended to include several templates and additional content in order to be able to generate a mailing. In addition, the event file required by UM is created in a further channel (preferably XML channel). The dispatch of the mailings can be triggered via a new workflow or a script. A script starts a schedule on the FirstSpirit Server, which generates the selected content and the event file and deploys these files. The UM recognises the new event file and then uses this file to work through the schedule.

The required scripts and two campaign templates are installed with the installation of the EmailMarketing module. Examples of the templates required to generate the above elements, as well as sections for the registration/deregistration and profile management for subscribers can be used from the enclosed example project.

2 Installing Universal Messenger

Apart from a FirstSpirit installation, Universal Messenger must also be installed for operation of the *EmailMarketing module*. This needs the following components:

- Database (for compatible databases, see UM documentation)
- JDBC drivers suitable for the selected database
- Apache Tomcat application server (is optionally installed by UM)
- SMTP server
- IMAP server

2.1 Installing Universal Messenger (Windows)

Installation of Universal Messenger Version 4.1 or higher is recommended. The Universal Messenger installation program guides you the installation and also preconfigures UM.

Of the components, you only need to select the Java Connector. This component contains a Java archive, which is necessary for the JSP pages for registering and cancelling the newsletter subscription.

For further configuration of the *EmailMarketing module,* please note the "open" password set during the installation of UM.

2.2 Installation under Linux

Unlike Windows, the installation under Linux usually takes place via the command line (shell), as most Linux systems do not have a graphic interface. The content of the configuration steps is identical to that of the Windows installation.

2.3 Operating Universal Messenger outside the Tomcat folder

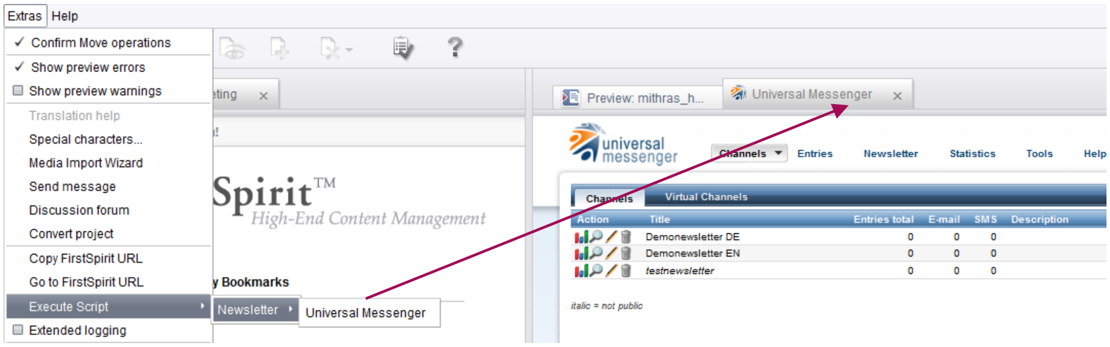
If Universal Messenger has been installed outside the webapps folder of the Tomcat server, the Tomcat should be configured for a new context. Either the paths to the Universal Messenger web application and to the cmsbs configuration file must be set in the server configuration (`tomcat/conf/server.xml`) or, alternatively, an XML with the same content (`conf\Catalina\localhost\cmsbs.xml`) must be created. The following example shows an appropriate configuration:

```
<Context path="/cmsbs" docBase="my/path/to/UM/cmsbs">

  <Parameter name="cmsbs.properties"

             value="my/path/to/UM/cmsbs-conf/cmsbs.properties" />

</Context>
```

3 Configuring Universal Messenger

After UM has been installed, it has to be configured so that mailings generated by the FirstSpirit Server can be automatically sent.

The configuration interface of UM is reached via the "Universal Messenger" menu script within the JavaClient. After calling the script, the interface opens in a new tab.



Action	Title	Entries total	E-mail	SMS	Description
	Demonewsletter DE	0	0	0	
	Demonewsletter EN	0	0	0	
	testnewsletter	0	0	0	

italic = not public

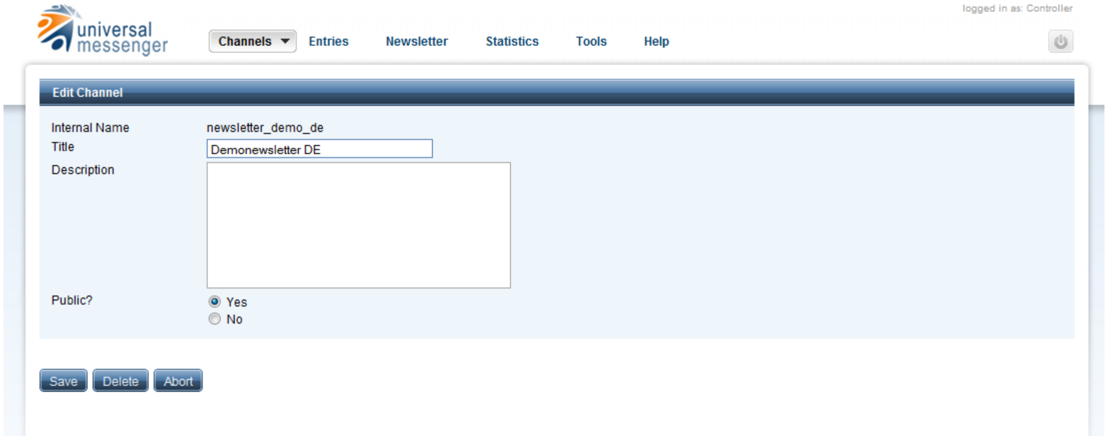
Alternatively, the configuration interface of UM can also be reached by calling the URL of Universal Messengers using your browser:

Example: http://localhost:8080/cmsbs

If the URL is not available, check the Java console of the Tomcat server and if necessary correct the licence and database configuration in the configuration file of Universal Messenger (`UM/cmsbs-conf/cmsbs.properties`).

3.1 Newsletter Channels

After logging in, the newsletter channels must be created. The channels in UM represent the different newsletters, for which a user can register on the website. In addition to the newsletter channels required, a channel must also be created for the test newsletter. This channel should not be public, as it is only used by the editors for test purposes and should not be subscribed by users. A new channel is created and edited using the first "Channels" menu item, in the top left hand side in the UM web application.



Edit Channel

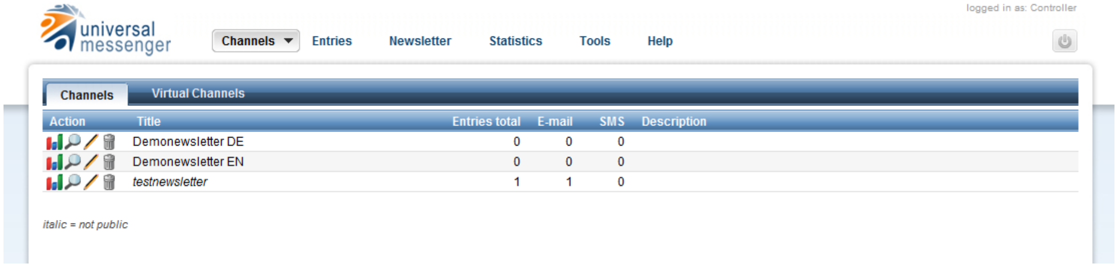
Internal Name newsletter_demo_de
Title Demonewsletter DE
Description
Public? ● Yes ○ No

Figure 3-1: Creating a channel

The internal name is used as a unique name, with which a channel can be identified. This name is required for the event file and must be entered in the relevant table in the FirstSpirit project.

The following channels must be created in order to prepare UM for the demonstration project supplied:

Testnewsletter	Internal name: testnewsletter Title: Test newsletter Public: no
Demo newsletter DE	Internal name: newsletter_demo_de Title: Demonewsletter DE Public: yes
Demo newsletter EN	Internal name: newsletter_demo_en Title: Demo newsletter EN Public: yes



Action	Title	Entries total	E-mail	SMS	Description
	Demonewsletter DE	0	0	0	
	Demonewsletter EN	0	0	0	
	testnewsletter	1	1	0	

italic = not public

Figure 3-2: List of the newsletter channels

Channels are not automatically synchronised between the FirstSpirit Server and UM, so that new channels have to be manually created in both systems.

3.2 Configure external link

Universal Messenger is able to accept and work through send jobs of external applications. To do this, Universal Messenger needs an event file, in which all information necessary for sending the mailing is given. This/these file(s) must be copied into a folder, which is monitored by Universal Messenger. If new event files are copied into this folder, this is registered by UM and the schedules/jobs are worked through on the basis of the event files.

If the FirstSpirit Server and Universal Messenger are not run on the same server, it is necessary to state a directory to which UM has direct read/write access. The event files generated by the FirstSpirit Server should be written into the directory via "Deployment Action" or using rsync.

The path of the monitored folder can be set in the Universal Messenger configuration file (*UM/cmsbs-conf/cmsbs.properties*) (see example).

```
cmsbs.directory.listen="pathToMyProj/newsletter/"
```

In Windows systems, ensure that a backslash "\" precedes all special characters. Here is an example:

```
cmsbs.directory.listen="C\:\\path\\to\\my\\proj\\newsletter\\"
```

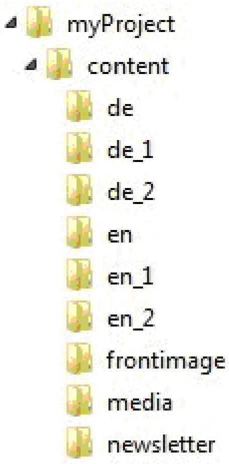
The newsletter module moves the generated event file directly into the generation directory in the "newsletter" folder, parallel to the generated output channels (de, de_1, …). If the generation directory is

```
.../myProject/content
```

the monitored folder must therefore be set to

```
.../myProject/content/newsletter/
```

(see Figure 3-3).



- myProject
  - content
    - de
    - de_1
    - de_2
    - en
    - en_1
    - en_2
    - frontimage
    - media
    - newsletter

Figure 3-3: Example of generated directory structure

It is necessary to configure an "open" password to use the http interface of Universal Messenger. This password is required to give FirstSpirit the opportunity to manipulate channels (test mail function)

```
cmsbs.open = "admin"
```

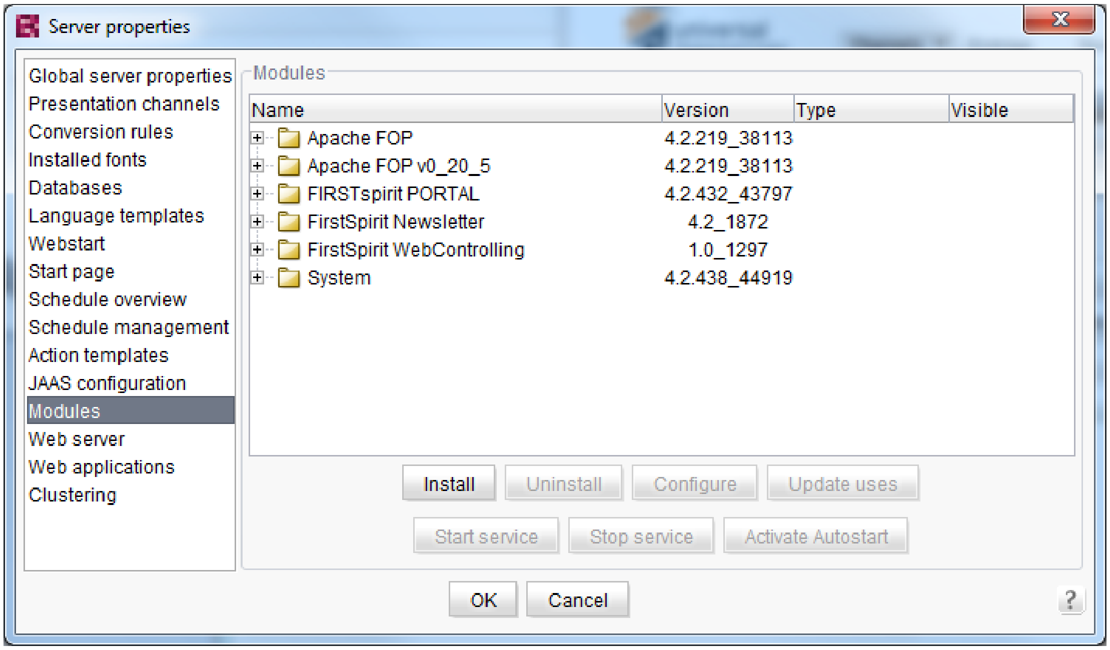
❗ *The stored password should not be the same as the UM Admin password. The password is always transferred in plain text.*

4 Installing and configuring the EmailMarketing module in FirstSpirit

The EmailMarketing module must be installed on the FirstSpirit Server with the help of the *fs-newsletter.fsm* file supplied. Furthermore, the module must be added to the project in order to make the necessary project-specific configuration.

4.1 Installation

To install the module, open the Server and Project Configuration. Open the server properties there and select the "Modules" area.

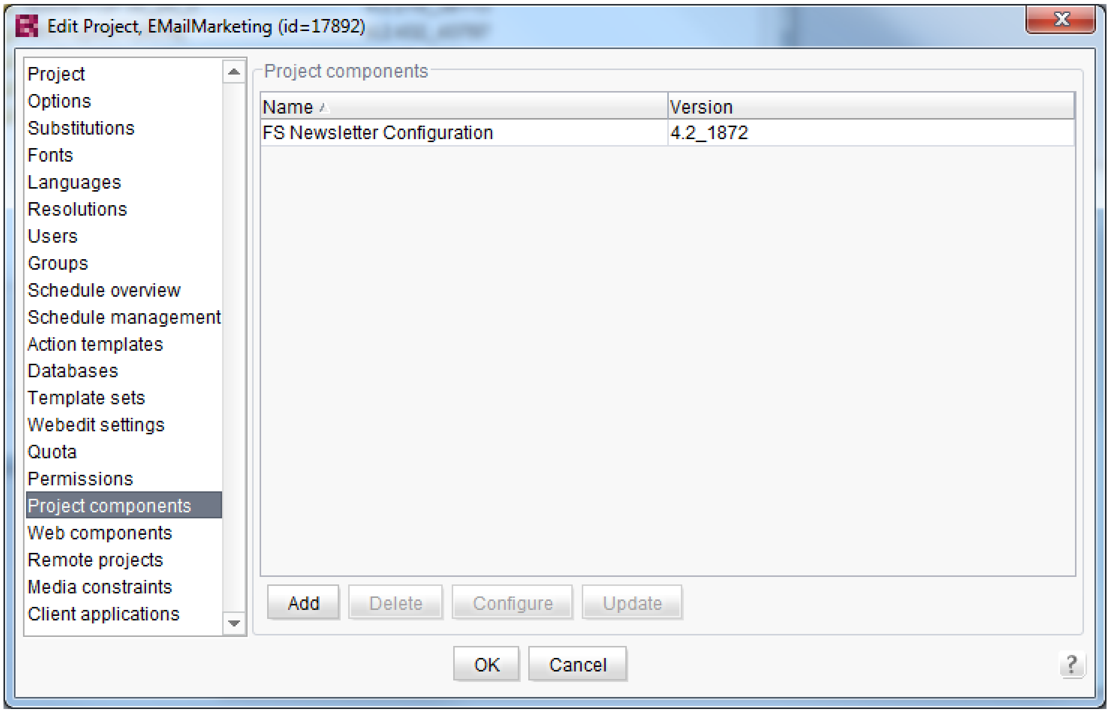


Name	Version	Type	Visible
Apache FOP	4.2.219_38113		
Apache FOP v0_20_5	4.2.219_38113		
FIRSTspirit PORTAL	4.2.432_43797		
FirstSpirit Newsletter	4.2_1872		
FirstSpirit WebControlling	1.0_1297		
System	4.2.438_44919		

Figure 4-1: Module list in the server properties

After clicking "Install", select the module file *fs-newsletter.fsm* and confirm with "Open". After the installation has finished, a new "FirstSpirit Newsletter" folder should have been added to the module overview. Now close the server properties by clicking "OK".

To enable project-specific configuration, the module must be added to the project which you would like to enhance with the module's functions. To do this, open the project configuration by double-clicking the relevant project in the project overview.



Name	Version
FS Newsletter Configuration	4.2_1872

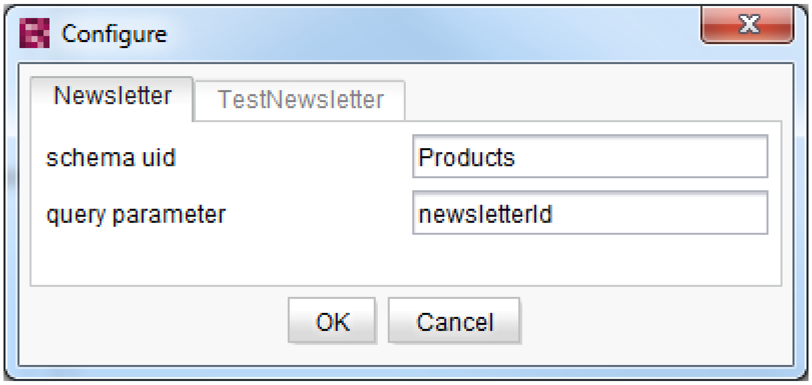
Figure 4-2: Project components in the project properties

On the left-hand side, select the "Project components" area. Now click "Add" and select the "FS Newsletter Configuration". The corresponding entry should now have been added to the list of project components.

4.2 Configuration

Double-click the "FS Newsletter Configuration" project component and the configuration dialog for the module opens. The following properties must be configured:

4.2.1 Newsletter



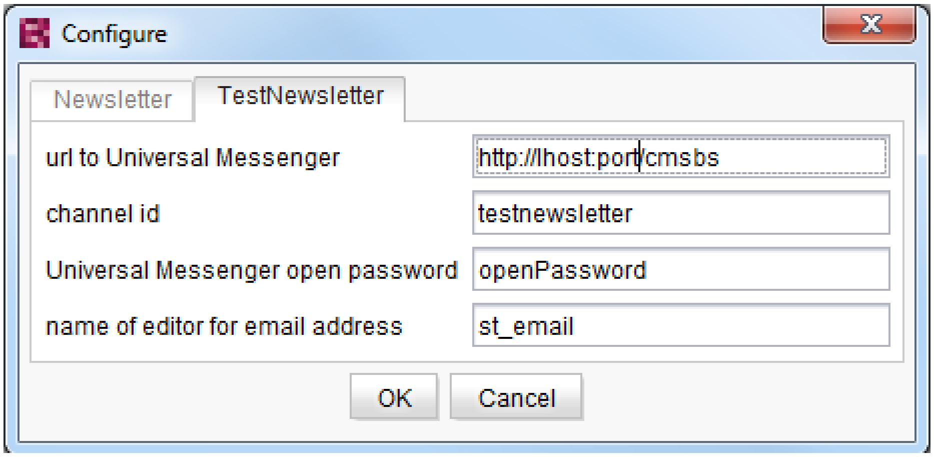
schema uid	Products
query parameter	newsletterId

Figure 4-3: "Newsletter" module configuration tab

schema uid
The reference name of the schema, in which the newsletter table is located.

query parameter
The scripts, which trigger the dispatch of the mailing, use a query, which limits the generation to the required dataset. To do this, the query parameter to be configured here is edited (see Chapter 5.6.3).

4.2.2 Test newsletter



url to Universal Messenger	http://lhost:port/cmsbs
channel id	testnewsletter
Universal Messenger open password	openPassword
name of editor for email address	st_email

Figure 4-4: "TestNewsletter" module configuration tab

url to Universal Messenger

The URL under which the Universal Manager web application can be reached must be given here.

channel id

The internal name of the channel to be used for the test mailing function is given here (see Chapter 3.1). This channel is managed by FirstSpirit and should not be manually filled with subscribers.

Universal Messenger open password

The test newsletter script needs the open password for UM. This password should have been configured as described in Chapter 3.2.

name of the editor for email address

The form definition of the test newsletter script contains an input component for the destination e-mail address. The name of this input component must be given here.

4.2.3 Using FirstSpirit 5.x

Using FirstSpirit version 5.x the following configuration parameters need to be added to the *server.xml* of the Apache Tomcat server.

<Connector port=”8080” useBodyEncodingForURI=”true” URIEncoding=”UTF-8”…/>

The parameters *useBodyEncodingForURI* and *URIEncoding* are essential in case of AdvancedURL usage for generation. They provide the UTF-8 encoding.

5 Extension of the project

In order to send mailings from a project, the project must be extended to include new elements. To enter new mailings, implementation with content sources was selected in the example project, but it is also possible to maintain via content pages. An existing database schema can be extended by two additional tables for maintenance in content sources. Alternatively, the schema from the example project including the table templates can be exported and imported as a new or further schema. In the first case, the table templates must be imported manually. To send the (test) mailing, the workflows can be added as an addition to the scripts. Two additional schedules are required in the project configuration, which generate the relevant mailing pages, perform a deployment and then move the event file into a directory provided for this purpose.

Pages for signing on and off at a channel/registering for a newsletter/cancelling a newsletter, as well as pages for editing the subscriber's profile data can be crated using appropriate section templates. Section templates with the relevant example code are provided in the example project supplied, in the Templates Store under *Section Templates* > *Newsletter.*

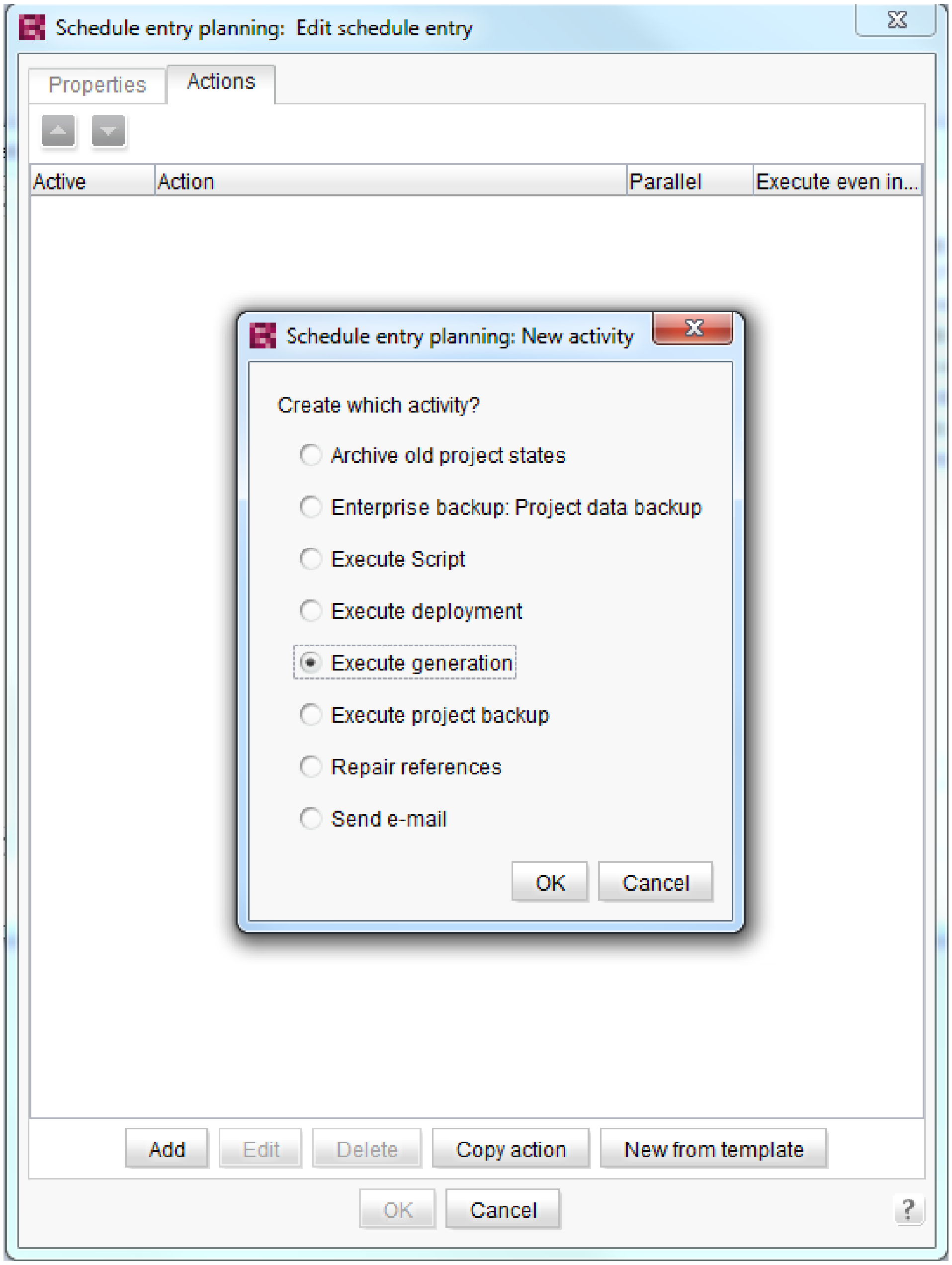
All the elements required are more precisely described in the following.

> ❗ *The demonstration project supplied already contains all the elements described here.*

5.1 Creating the server schedules

Two new schedules must be created in the schedule management: one for sending the real mailing and one for sending the test mailing. The difference between the two schedules is that no statistics data is written back in the test mailing schedule, i.e. unlike the schedule of the real mailing, the "Statistic" action is omitted.

5.1.1 Schedule Newsletter

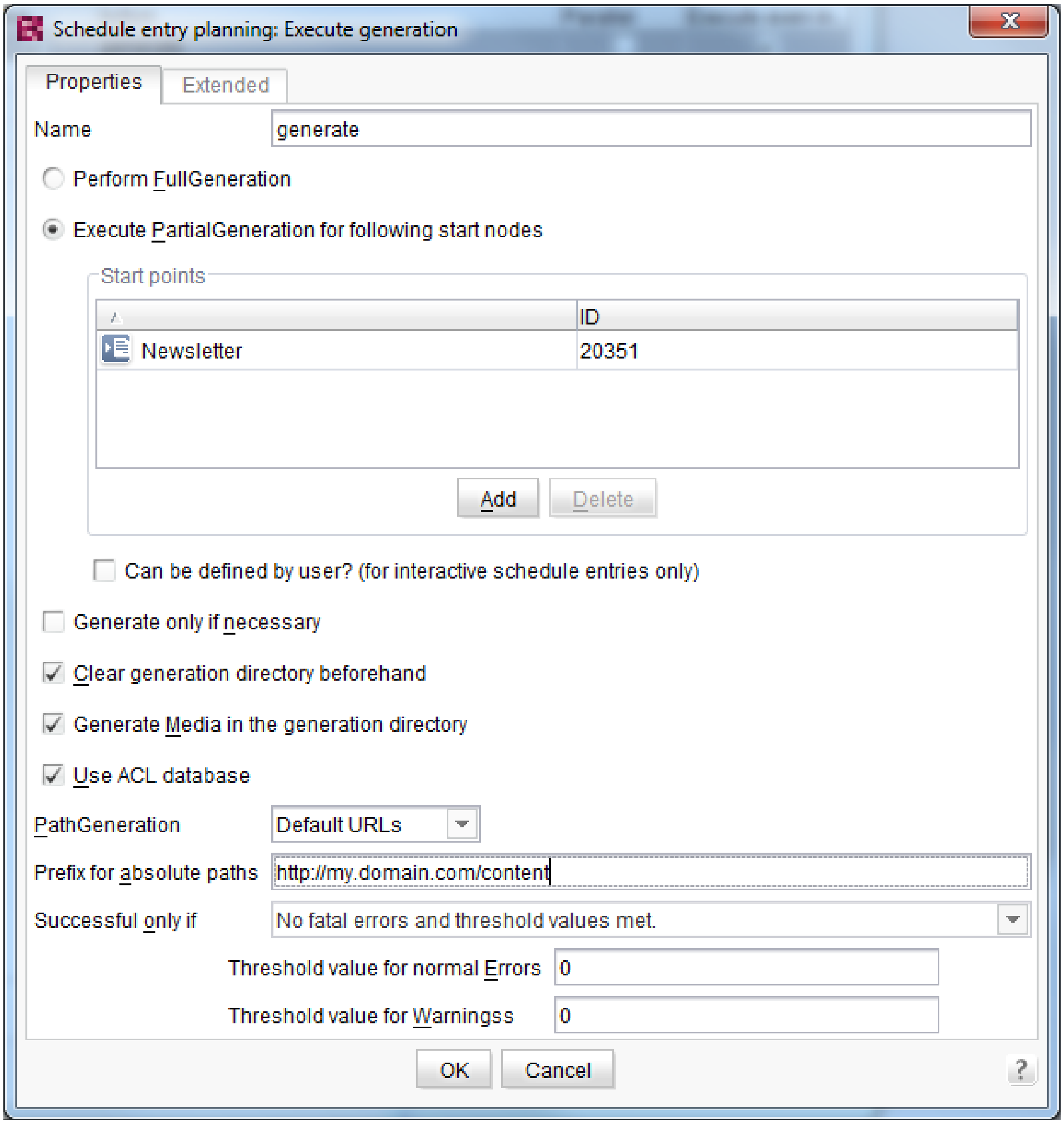


Create which activity?

○ Archive old project states
○ Enterprise backup: Project data backup
○ Execute Script
○ Execute deployment
● Execute generation
○ Execute project backup
○ Repair references
○ Send e-mail

Figure 5-1: Add generation action

In the project-specific schedule management, create a new schedule and first of all, add a generation action.



Name: generate

○ Perform FullGeneration

● Execute PartialGeneration for following start nodes

	ID
Newsletter	20351

□ Can be defined by user? (for interactive schedule entries only)

□ Generate only if necessary

☑ Clear generation directory beforehand

☑ Generate Media in the generation directory

☑ Use ACL database

PathGeneration: Default URLs

Prefix for absolute paths: http://my.domain.com/content

Successful only if: No fatal errors and threshold values met.

Threshold value for normal Errors: 0

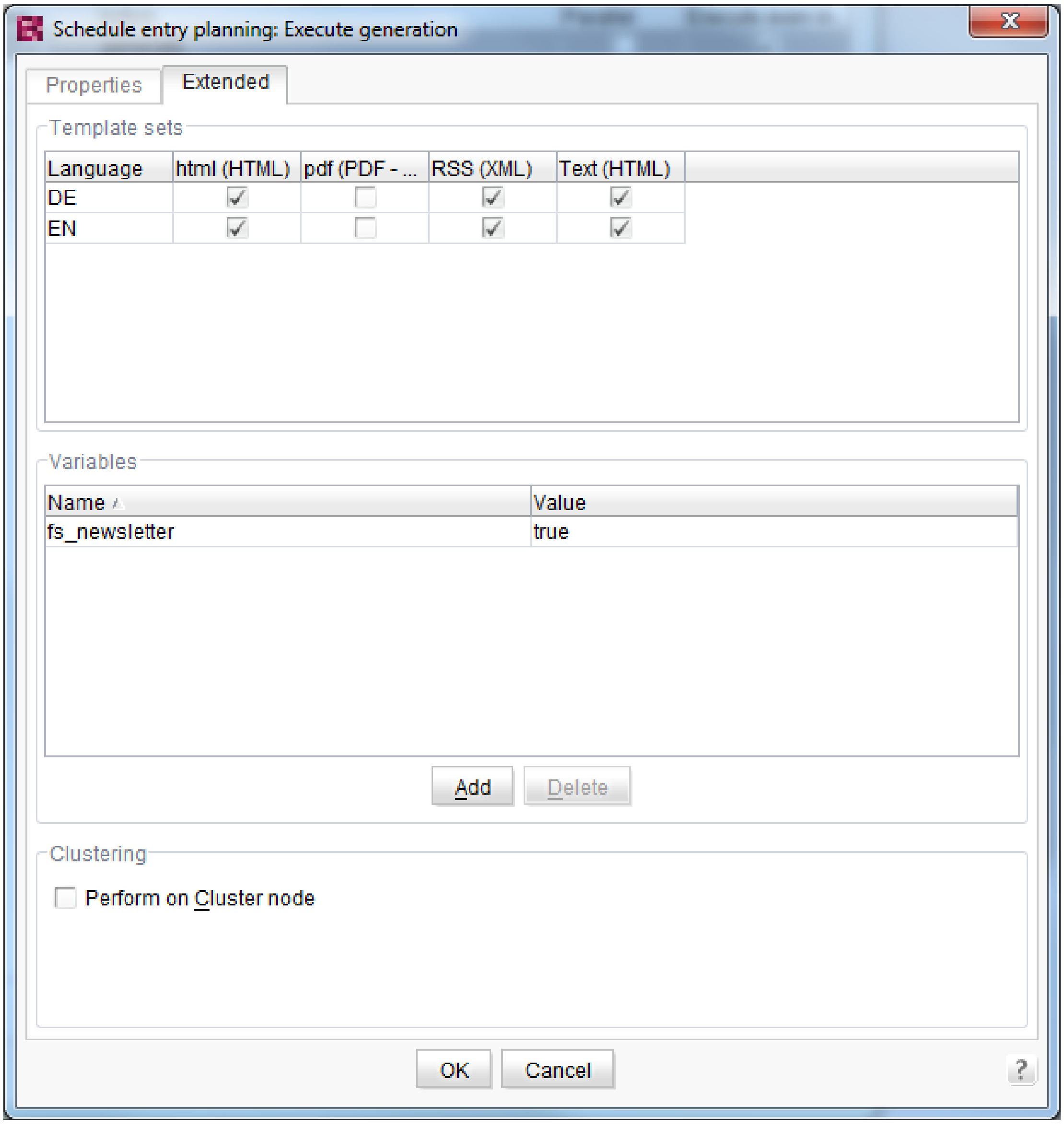
Threshold value for Warningss: 0

Figure 5-2: Properties of the generation

Select the partial generation and add the mailing reference to the list of the starting nodes.

Activate the "Clear generation directory beforehand" checkbox.

Fill the "Prefix for absolute paths field, suitably for your project.



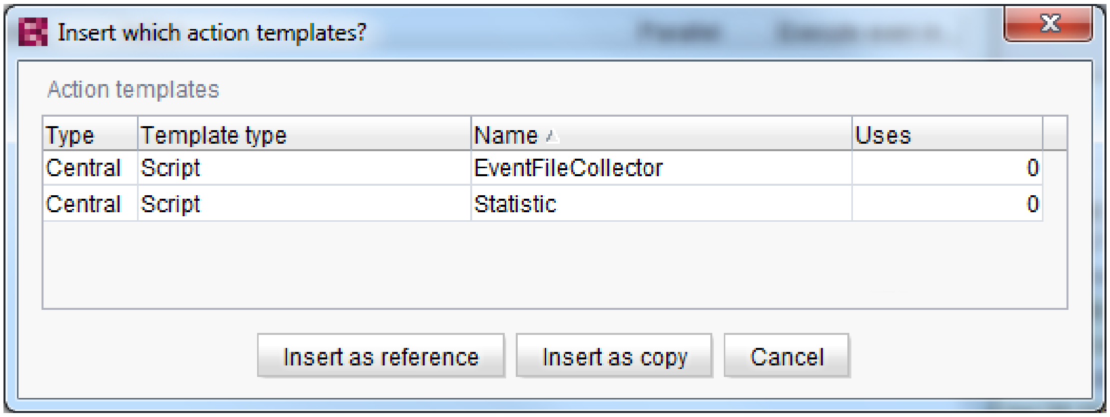
Language	html (HTML)	pdf (PDF - ...	RSS (XML)	Text (HTML)
DE	☑	☐	☑	☑
EN	☑	☐	☑	☑

Name	Value
fs_newsletter	true

☐ Perform on Cluster node

Figure 5-3: Extended settings of the generation action

On the "Extended" tab, select the languages and output channels to be generated. Ensure that the output channels are also selected.



Insert which action templates?

Action templates

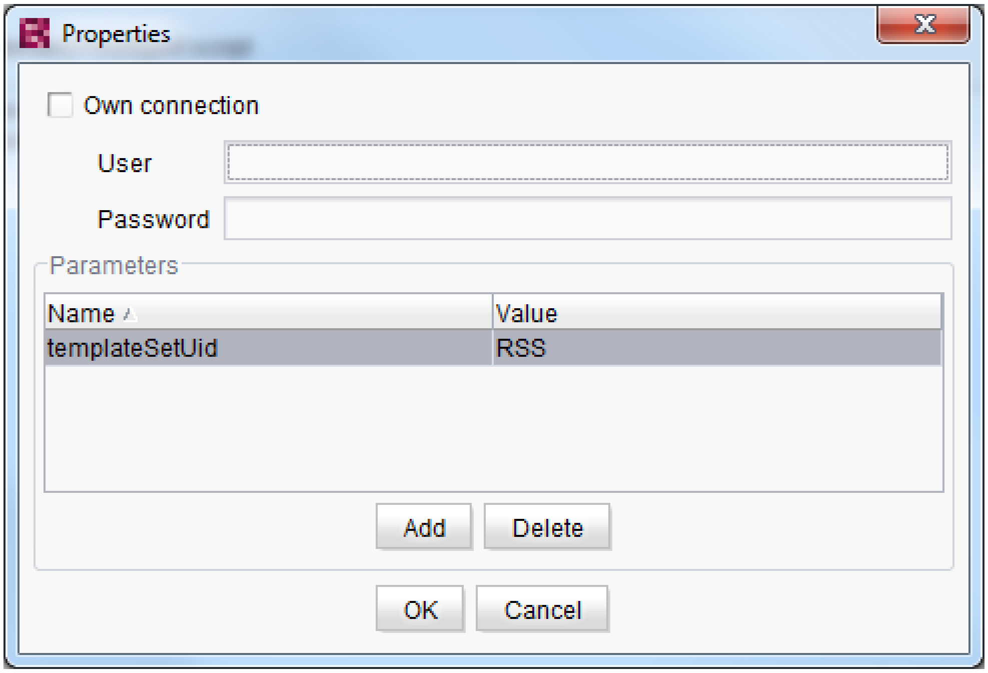
Type	Template type	Name	Uses
Central	Script	EventFileCollector	0
Central	Script	Statistic	0

Figure 5-4: Selection of the action template

Confirm the settings by clicking "OK".

Next, create an action by clicking "New from template".

When the module was installed, two action templates were installed on the server: "EventFileCollector" and "Statistic". Select "EventFileCollector" and add the action as a copy.



Properties

□ Own connection

User

Password

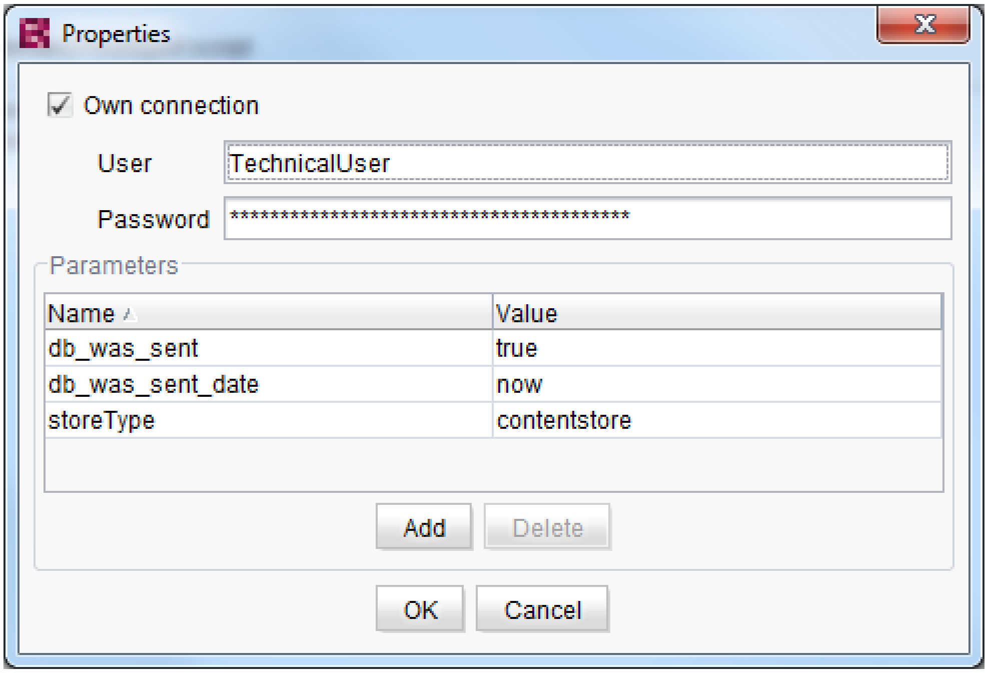
Parameters

Name	Value
templateSetUid	RSS

Figure 5-5: Properties of the EventFileCollector action

Open the properties of the action with a double-click and add the "templateSetUid" parameter. This parameter must be set to the name of the channel used to generate the event file, As the third schedule action, create another copy of an action template. This time, however, the action template is "Statistic", which should also be added as a copy.

A separate connection must be used for this action, as this action must access the project data. Therefore, it is useful to select a technical user who has write permissions for the relevant data. We advise against using the ServerAdmin account.



☑ Own connection

User TechnicalUser

Password **

Parameters

Name	Value
db_was_sent	true
db_was_sent_date	now
storeType	contentstore

Figure 5-6: Properties of the "Statistic" action

The statistic action can write back into the dataset after sending the mailing. These values can be given here in the parameter list. In the demo project supplied, the following values are written back:

- **db_was_sent**
  This parameter is used to save the information in the dataset that the mailing has been sent. This value can be used to prevent renewed sending.

- **db_was_sent_date**
  This parameter shows when the mailing was sent. "now" is used as a wildcard for the time at which the action was performed.

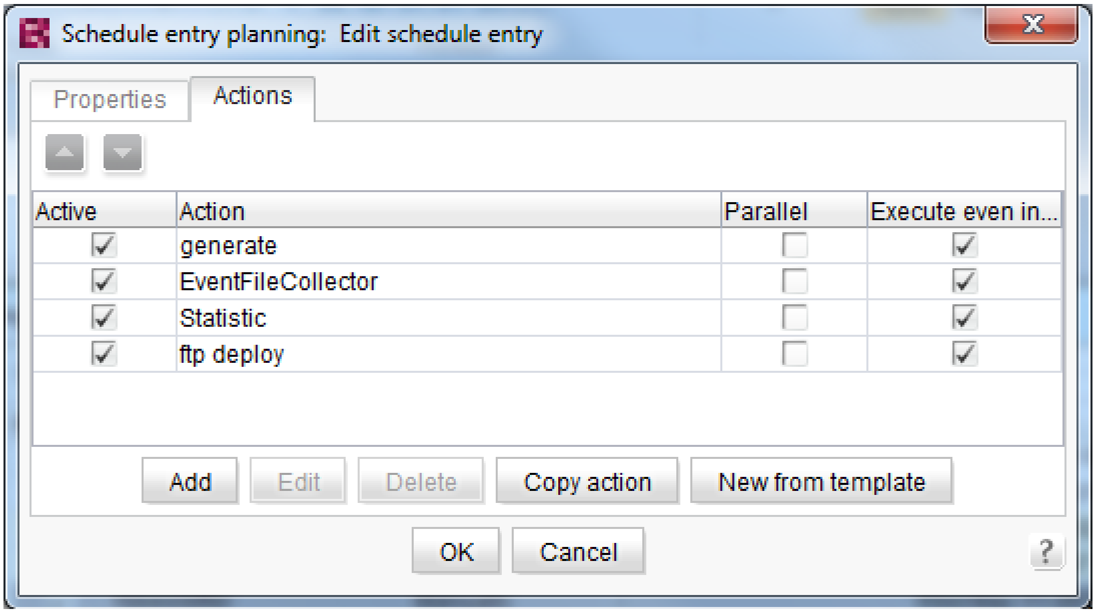
The values to be set are freely configurable and are used according to the schema *ParameterName = unique identifier of the input component* and *ParameterValue = value of the input component*. The value to be adopted must, of course, be set according to the input component.

After adding the action, the "storeType" parameter is already created. This parameter is used to give the store in which the mailings are maintained.

- **storeType**
  Gives the store in which the mailings are maintained. Valid values for this parameter are "contentstore" or "pagestore".

A deployment action must be added as a further action.

The following figure shows the list of actions of a finished schedule taken from the demo project supplied.

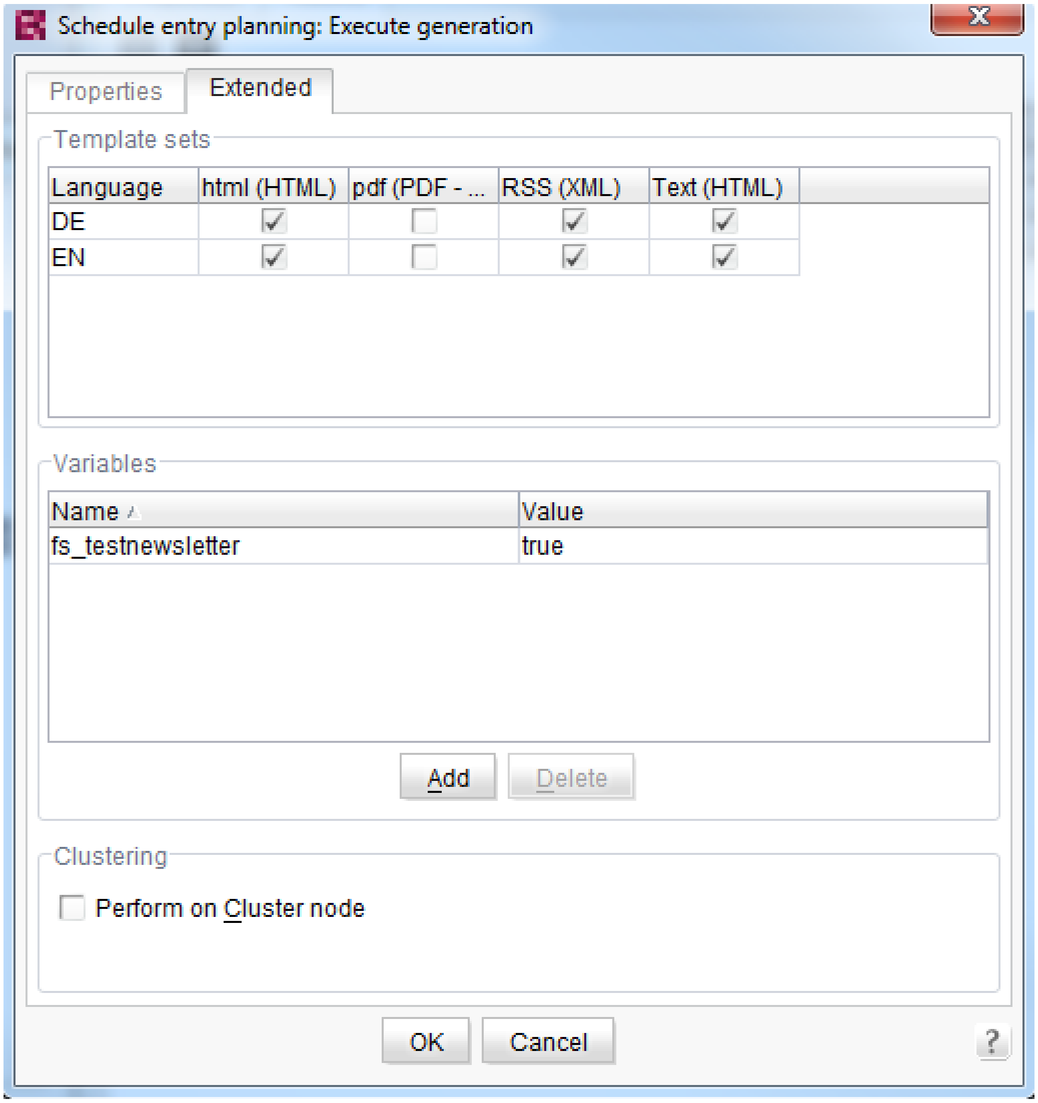


Active	Action	Parallel	Execute even in...
☑	generate	☐	☑
☑	EventFileCollector	☐	☑
☑	Statistic	☐	☑
☑	ftp deploy	☐	☑

Figure 5-7: Newsletter schedule of the demo project

5.1.2 Schedule test newsletter

A schedule must also be created for the test newsletter. This must initially also contain a generation action (see Chapter 5.1.1).). The only difference is that the variable "fs_testnewsletter" must be set with the value "true" in the variables list in the advanced settings of the generation action.



Template sets

Language	html (HTML)	pdf (PDF - ...	RSS (XML)	Text (HTML)
DE	☑	□	☑	☑
EN	☑	□	☑	☑

Variables

Name	Value
fs_testnewsletter	true

Clustering

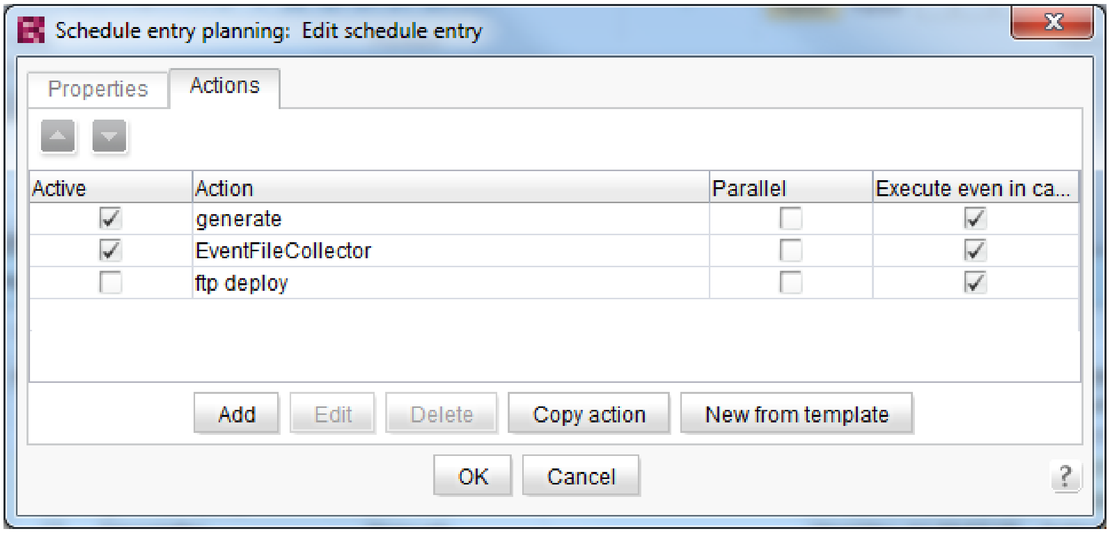
□ Perform on Cluster node

Figure 5-8: Extended properties of the generation action

Furthermore, as in the newsletter schedule, the "EventFileCollector" action is required. To create this action, follow the description in Chapter 5.1.1.

The third action is the deployment. You can create this adjusted to your project.

The following figure shows the list of actions of the finished schedule taken from the demo project supplied.



Active	Action	Parallel	Execute even in ca...
☑	generate	□	☑
☑	EventFileCollector	□	☑
□	ftp deploy	□	☑

Figure 5-9: Test newsletter schedule of the demo project

5.2 Adjusting the workflow scripts

When the project component was installed, the "Send newsletter" and "Send test newsletter" scripts were added to your project. These scripts still have to be adjusted to the current project.

The scripts each contain the following parameter:

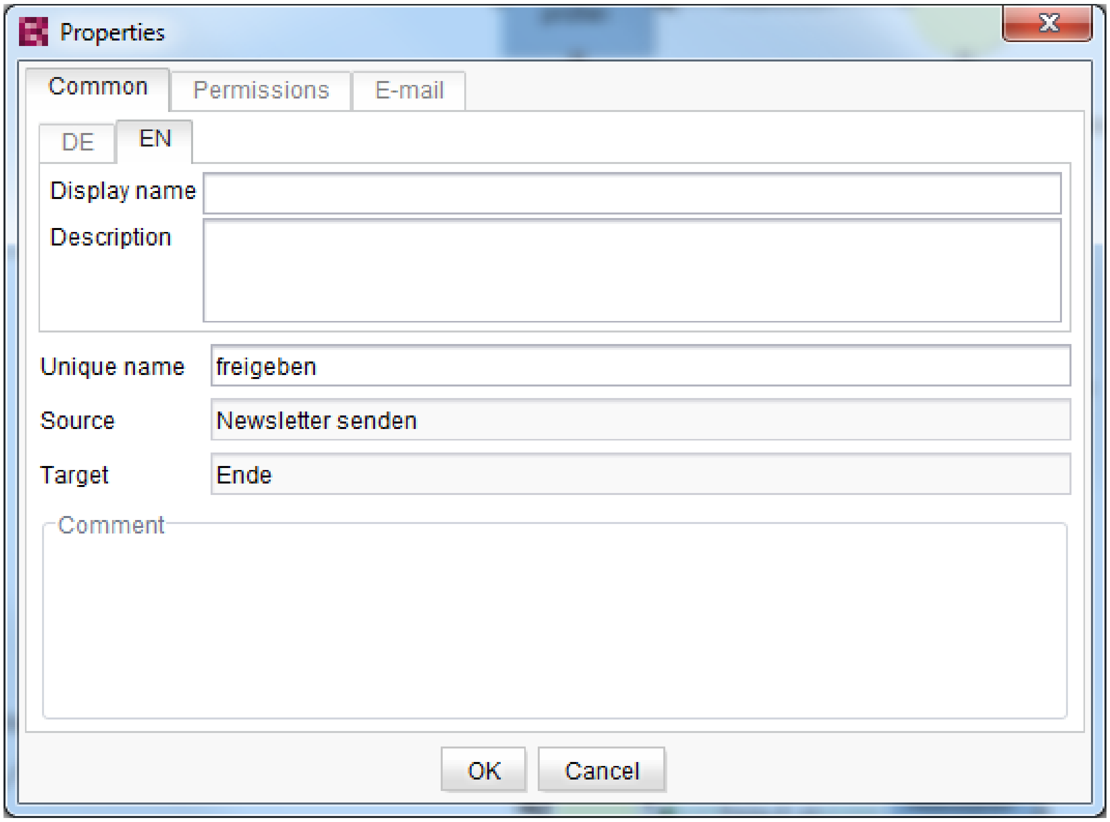
```
// uid of newsletter pageref
String pageRefUid = "newsletter_1";
```

This parameter must be adjusted as follows.

`pageRefUid`	This variable must contain the unique reference name of the page reference of the mailing in the Site Store. This value must be the same in both scripts.

If the scripts are started from a workflow, another parameter must be handed over to the "sendNewsletter(…)" method. If the script has been successfully run and the schedule has been finished, a transition is switched, which can be defined using the "targetTrans" parameter.

`targetTrans` Defines the transition, which is switched following successful execution of the script. This parameter must match the unique name of the transition from the workflow.



Unique name: freigeben

Source: Newsletter senden

Target: Ende

Figure 5-10: Properties of the transmission

If the script is to be started without workflow, directly from the context menu, this parameter can be omitted from the function call. Example:

```
import de.espirit.firstspirit.ps.newsletter.Newsletter;

Newsletter.sendNewsletter(context,scheduleId, pageRefUid);
```

5.3 Creating the workflows

The (test) dispatch is controlled by the editor using two workflows. These are not imported into the project when the module is installed and must be suitably created on your project. The following figure shows the workflow for sending the newsletter from the demo project.

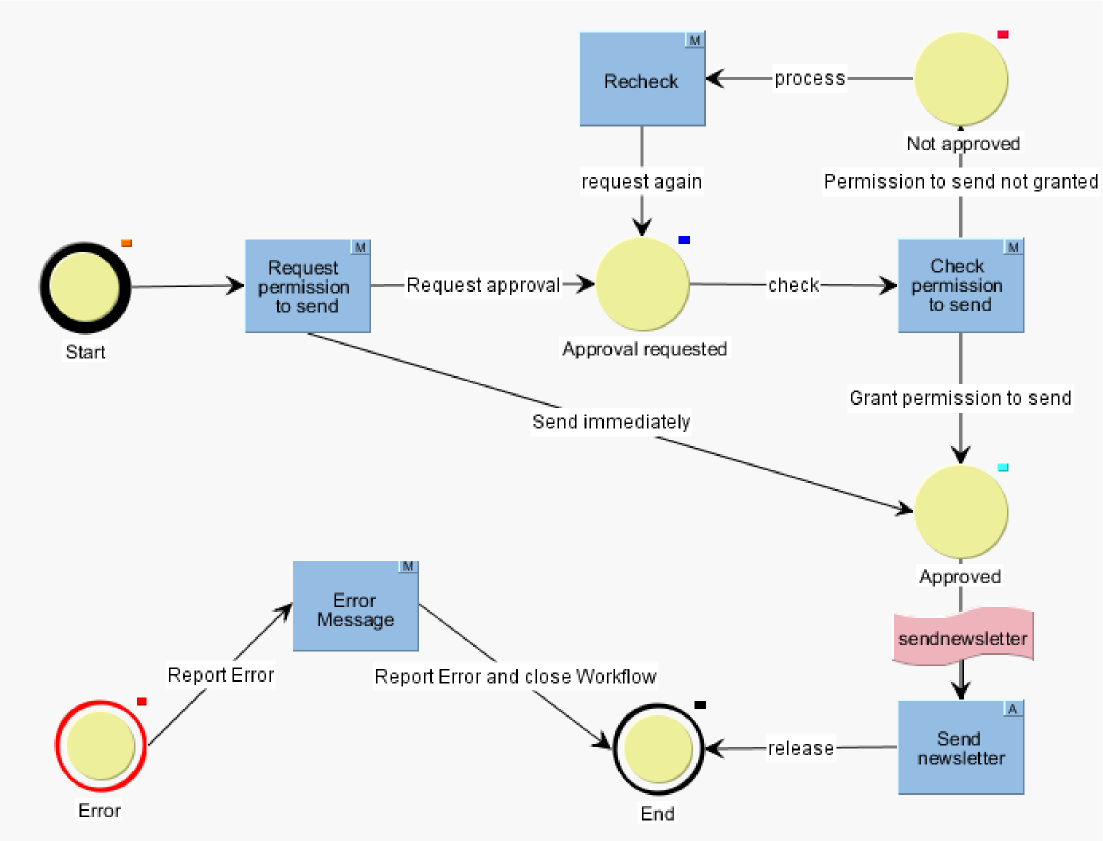


Figure 5-11: Example workflow for sending the newsletter

The view logic of the workflows ensures that they can only be performed on released datasets.

In order for the workflow to start the dispatch, the "Send newsletter" script must be added to an activity. This script was imported with the installation in your project.

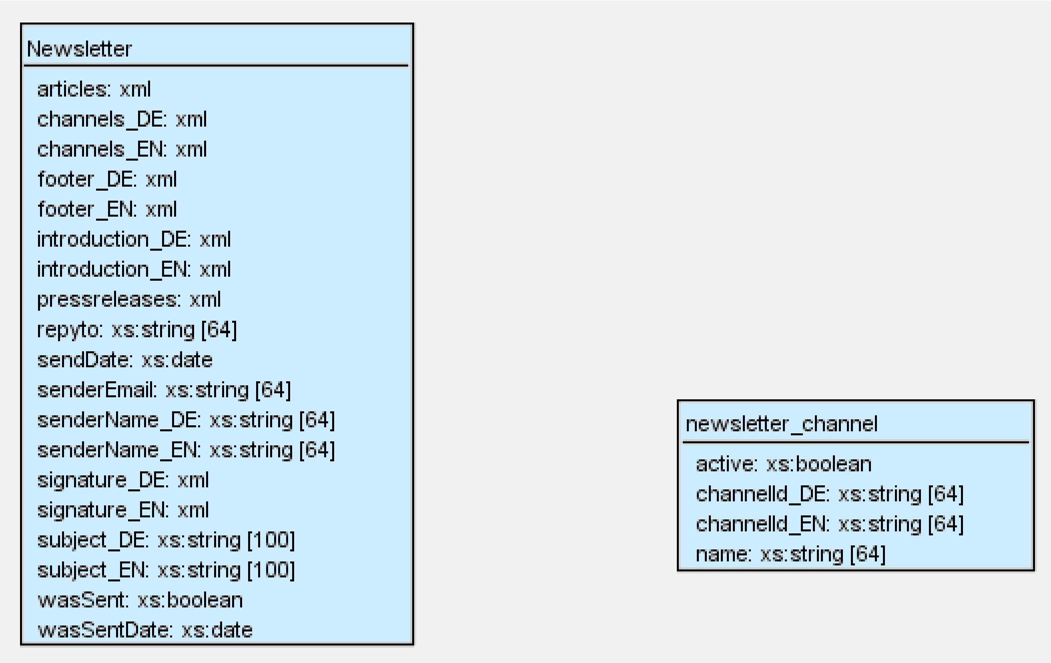
5.4 Using newsletter scripts by context menu

The delivered newsletter scripts may only be executed by users who hold administrative permissions.

Before executing the scripts they have to be configured in order to be accessible via context menu. Particularly the visibility of the scripts needs to be adapted and furthermore a control structure that differs between workflow and context menu has to be implemented.

5.5 Adjust database schema

A table for the mailings and a table for the channels must be created in the database schema. Both tables must be linked with an M:N relationship.



Newsletter

articles: xml
channels_DE: xml
channels_EN: xml
footer_DE: xml
footer_EN: xml
introduction_DE: xml
introduction_EN: xml
pressreleases: xml
repyto: xs:string [64]
sendDate: xs:date
senderEmail: xs:string [64]
senderName_DE: xs:string [64]
senderName_EN: xs:string [64]
signature_DE: xml
signature_EN: xml
subject_DE: xs:string [100]
subject_EN: xs:string [100]
wasSent: xs:boolean
wasSentDate: xs:date

newsletter_channel

active: xs:boolean
channelId_DE: xs:string [64]
channelId_EN: xs:string [64]
name: xs:string [64]

Figure 5-12: The newsletter tables

5.5.1 Newsletter/(Mailing) table

The newsletter table in the demo project supplied contains the following columns:

- **articles** (xml, language-independent)
  Among other things, the content of the newsletter consists of article sections.

- **footer** (cms, language-dependent)
  Footnote, displayed at the end of the newsletter.

- **introduction** (xml, language-dependent)
  Introductory text of the newsletter.

- **pressreleases** (xml, language-independent)
  The second part of the newsletter content consists of press releases. The press releases are copied from the "Press releases“ table.

- **replyto** (string[64], language-independent)
  Reply to-e-mail address of the newsletter.

- **sendDate** (date, language-independent)
  Date on which the newsletter is to be sent. If this field is left empty, the newsletter is sent immediately.

- **senderEmail** (string[64], language-independent)
  E-mail address of the sender.

- **senderName** (string[64], language-dependent)
  Name of the sender.

- **signature** (xml, language-dependent)
  Signature of the newsletter

- **subject** (xml, language-dependent)
  Newsletter subject

- **wasSent** (boolean, language-independent)
  States whether the newsletter has already been sent or not. This value is only set by the schedule script, which is executed by the schedule following deployment of the newsletter.

- **wasSentDate** (date, language-independent)
  Gives the data on which the newsletter was sent. This value is only set by the schedule script, which is executed by the schedule following deployment of the newsletter.

5.5.2 channel table

The newsletter channel table in the demo project supplied contains the following columns:

- **active** (boolean, language-independent)
  This value can be used to deactivate a newsletter channel so that newsletters cannot be sent via this channel.

- **name** (string [64], language-independent)
  Name of the channel.

- **channelId** (string [64], language-independent)
  The unique name of the channel from UM is saved here. The unique name of a channel can be read in the channel configuration in UM.

5.6 Templates

At least two table templates and one page template are required in order to be able to generate all the necessary pages required for the dispatch. The implementation from the demo project is described in the following.

5.6.1 Table template for the newsletter

The "newsletter" table template for the newsletter table is used to define the form for the input of datasets, as well as the output of the HTML channel, which represents the actual mailing. The event file is generated in another channel (here: XML) of this table template.

A concealed data input component with current date and time ensures that all changes to the dataset generate a new revision of the dataset, so that the most recent data is always taken into account in the downstream deployment.

The following gives an example for a generated XML event file. This file contains information about

- the send date,
- the channels to be used,
- the sender e-mail address,
- the subject of the mailing and
- the content of the mailing, each in text and in HTML format

```
<?xml version="1.0" encoding="UTF-8"?>
<event>
  <date format="dd.MM.yyyy HH:mm:ss">12.04.2011 15:08:43</date>
  <destination>
    <channel>newslettertest</channel>
  </destination>
  <data>
    <email sender="newsletter@fs4demo.e-spirit.de"
         replyto="newsletter@fs4demo.e-spirit.de">
      <subject>First edition</subject>
      <plaintext inline="false">
        /path/to/content.txt
      </plaintext>
      <htmltext inline="false" embedImages="all">
        /path/to/content.html
      </htmltext>
    </email>
  </data>
</event>
```

For further information about the XML event file, refer to the UM documentation for developers.

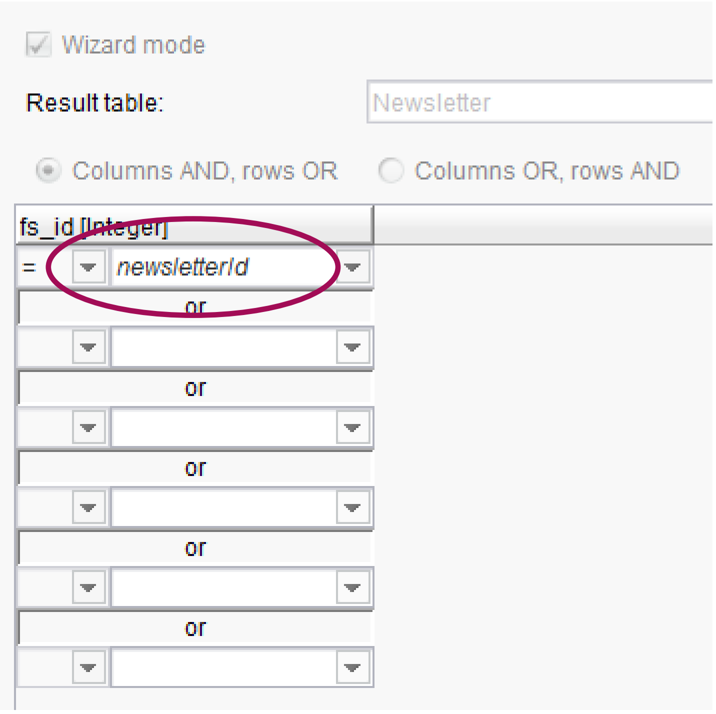
5.6.2 Table templates for newsletter channels

The table template for the newsletter channels is required in order to define the input form for the content source. The output channels can remain empty.

5.6.3 Database query

A query is required, with the help of which the selected dataset from the newsletter content source is selected so that the system only generates the single dataset required. The query parameter is automatically filled with the ID of the selected dataset.

This parameter name must be entered in the project-specific configuration of the module (see Chapter 4.2.1).



☑ Wizard mode

Result table: Newsletter

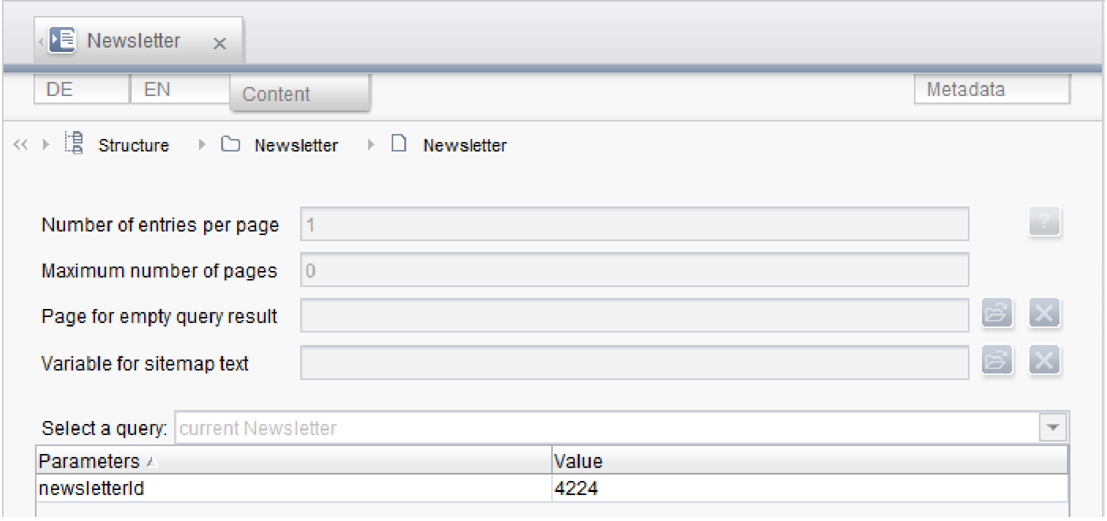
● Columns AND, rows OR ○ Columns OR, rows AND

fs_id [Integer]

= newsletterId

Figure 5-13: Query parameter for the newsletter dataset

The new query created here must be selected on the "Data" tab of the page reference used for content projection. The query parameter (here: "newsletterId") does not have to be set here, as this is done automatically.



Number of entries per page: 1

Maximum number of pages: 0

Page for empty query result:

Variable for sitemap text:

Select a query: current Newsletter

Parameters	Value
newsletterId	4224

Figure 5-14: Data tab of the page reference of the newsletter page

5.7 Project settings

In the demo project there is a project settings page in which, among other things, settings can be made for UM.

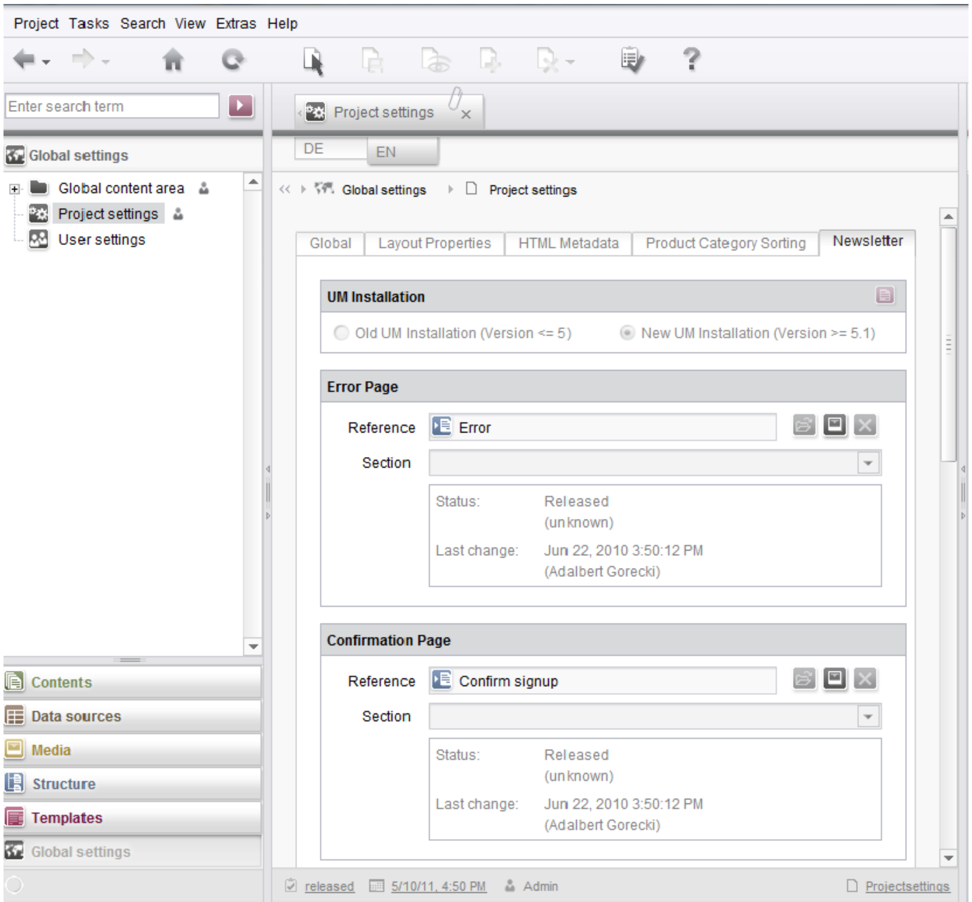
First, the installed version of UM can be selected.

The first option: "Old UM installation (Version <= 5)" should be selected if the installed UM version <= Version 5 or an upgrade has been made from a version <= Version 5 to, e.g. version 5.2 (default).

The second option: "New UM installation (Version >= 5.1)" should be selected if a new installation has been performed with a version >= Version 5.1.

If you would like to run a new installation of UM of one version >= Version 5.1 with the data of an old UM version, please contact the support department of Pinuts GmbH, as this requires further adjustments.

Other settings options that can be set are the references to various pages such as an errors page, confirmation page, etc., which are required by UM for linking.



UM Installation

○ Old UM Installation (Version <= 5)   ● New UM Installation (Version >= 5.1)

Error Page

Reference: Error

Section:

Status: Released (unknown)

Last change: Jun 22, 2010 3:50:12 PM (Adalbert Gorecki)

Confirmation Page

Reference: Confirm signup

Section:

Status: Released (unknown)

Last change: Jun 22, 2010 3:50:12 PM (Adalbert Gorecki)

6 JSP pages

The following chapter describes the templates of the example project used. The templates were used to implement a double opt-in/out as well as subscriber profile management.

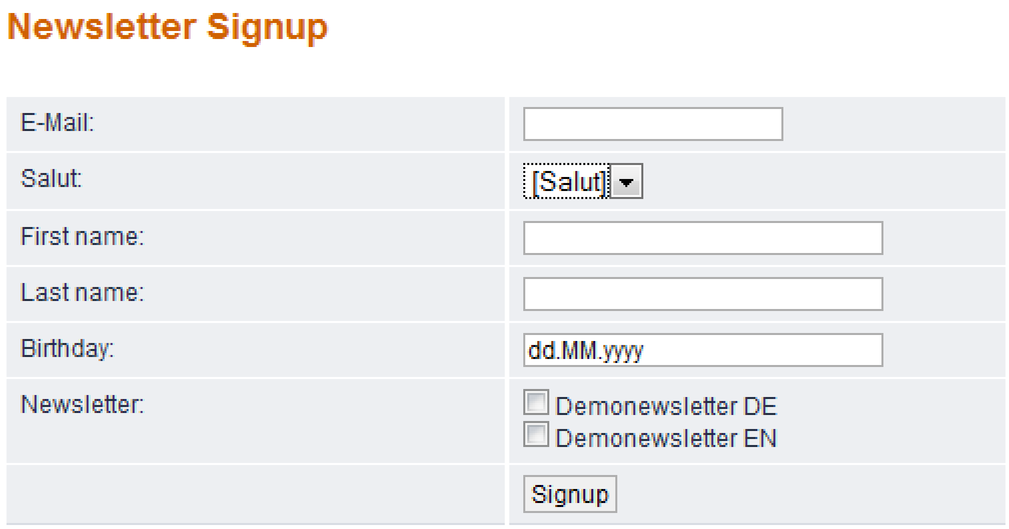
The JSP pages with which the user can e.g. register for a newsletter on the website, are created using the standard page template of the demo project. The individual functions (register, login, edit profile …) are organised into individual section templates. The demo project supplied contains section templates for the following functions:

- Register for newsletter
- Confirm registration
- Cancel newsletter
- Login
- Edit profile
- Password forgotten

These section templates are used in the demo project by way of example and demo pages are generated for the registration/deregistration or signing on/signing off, login, "Password forgotten" and "Edit profile".

6.1 Register for newsletter

Registration of a user via the website is implemented on the following page:



Newsletter Signup

E-Mail:	
Salut:	[Salut] ▾
First name:	
Last name:	
Birthday:	dd.MM.yyyy
Newsletter:	□ Demonewsletter DE □ Demonewsletter EN
	Signup

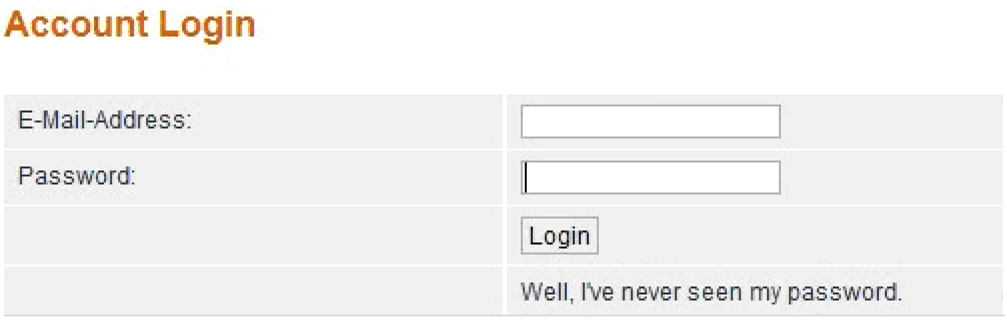
Figure 6-1: Newsletter registration

After registering (signing on) for the newsletter, the user then receives a confirmation e-mail including a link, which completes the registration (double opt-in).

The JSP example code for such a section template, as well as for the confirmation page can be found in the "Newsletter signup" and "Newsletter confirm" section templates.

6.2 Edit user data

The user must log-in in order to edit their data, password and their subscription.

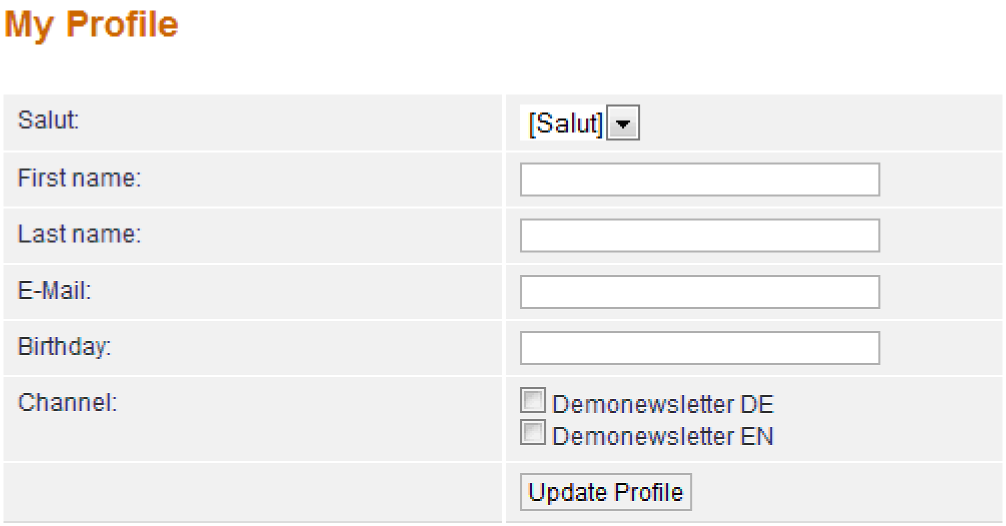


Account Login

E-Mail-Address:	
Password:	
	Login
	Well, I've never seen my password.

Figure 6-2: The registration mask

Next, the user can use the following mask (screen form) to change their personal data:



My Profile

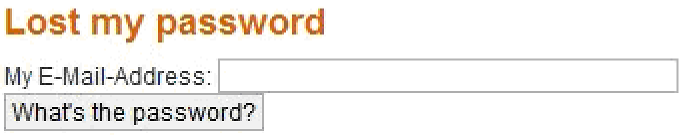
Salut:	[Salut] ▾
First name:	
Last name:	
E-Mail:	
Birthday:	
Channel:	□ Demonewsletter DE □ Demonewsletter EN
	Update Profile

Figure 6-3: Personal data

The JSP example code for these section templates can be found in the example project in the "Newsletter login" and "Newsletter edit profile" section templates.

6.3 Password forgotten

If the user has forgotten their password, they can use the following mask to have it sent to them again:



Lost my password

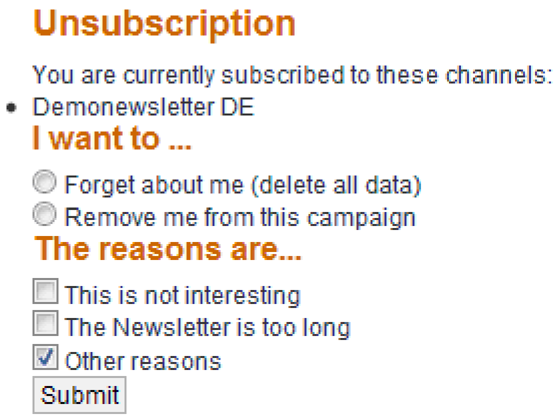
My E-Mail-Address:

Figure 6-4: Password forgotten

The JSP example code for such a section template can be found in the “Newsletter password forgotten” section template in the example project.

6.4 Cancel the newsletter

Another page is required so that a subscriber can cancel the newsletter. This page signs off the user in UM from the relevant channel (cancels the newsletter subscription) and deletes their data, if they wish.



Unsubscription

You are currently subscribed to these channels:

- Demonewsletter DE

I want to ...

○ Forget about me (delete all data)
○ Remove me from this campaign

The reasons are...

□ This is not interesting
□ The Newsletter is too long
☑ Other reasons

Figure 6-5: Cancel newsletter subscription

The JSP example code for such a section template can be found in the example project in the "Newsletter unsubscribe " section template.

The link to the page for cancelling the newsletter is integrated in the table template of the newsletter. This link has the following structure:

```
<a
href="http://my.domain.com/mynewsletter/unsubscribe.jsp?c={msgid}">Unsubscri
be from newsletter</a>
```

If the newsletter is sent by UM, it replaces the "msgid" parameter with a user ID, required for the cancellation page, in order to identify the user.

The example code for this link is given in the example project in the newsletter table template.

6.5 Configuring the web application

The JSP pages, which are generated with the help of the section templates described above, require a range of Jar files for correct execution. These files are in the */UM/java/lib* directory. Copy all files from the directory into the */WEB-INF/lib/* directory of your deployed web application. (It may be necessary to create this directory first.)

In addition, the following context parameter must be set in the *web.xml* of the web application of the deployed pages, analogous to the *web.xml* in the *cmsbs-client.war*:

```
<context-param>
  <param-name>cmsbs.umurl</param-name>
  <param-value>http://domain:port/cmsbs/UM</param-value>
</context-param>
<context-param>
  <param-name>cmsbs.open</param-name>
  <param-value>open.password</param-value>
</context-param>
```

`cmsbs.umurl` This parameter gives the URL to the installed instance of UM.

`cmsbs.open` The open password of UM. This password is given in the UM configuration file. The UM configuration file is located under */pathToMy/UM/cmsbs-conf/cmsbs.properties*.

7 Send mailings with FirstSpirit

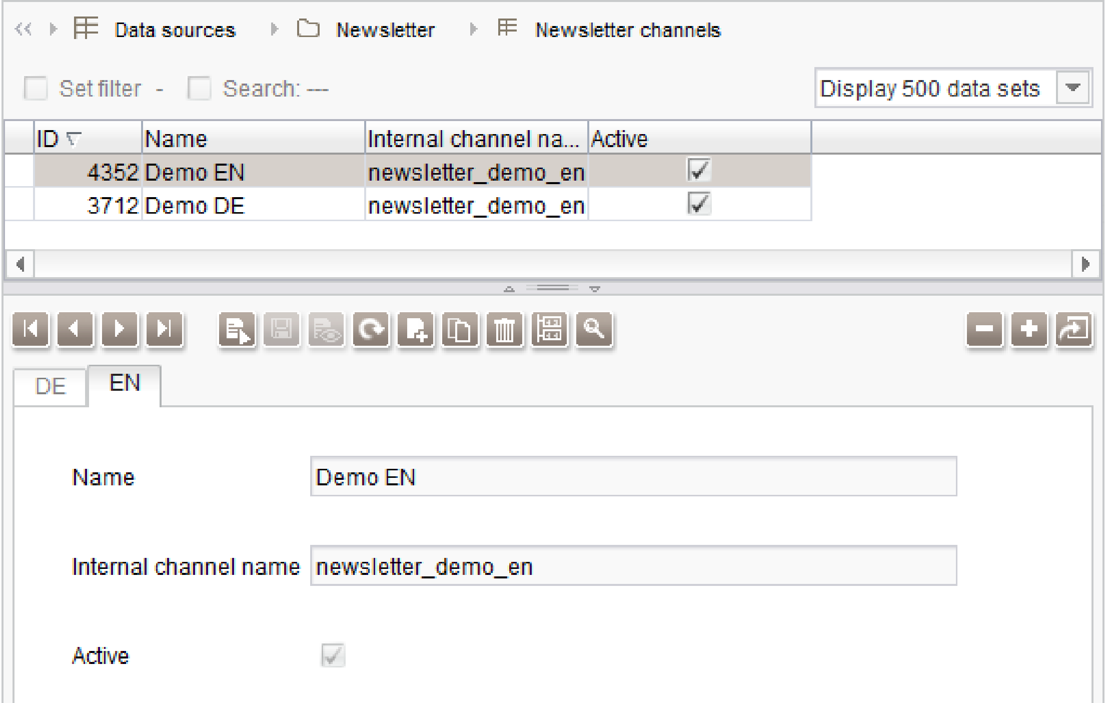
This chapter describes how a mailing can be sent from FirstSpirit and which datasets are necessary.

7.1 Datasets required

The following section shows, using the example of the example project supplied, which datasets are required to send a mailing.

To send a newsletter in the example project, at least two datasets are required: on the one hand the dataset for the newsletter, on the other the dataset for the newsletter channel. You can use the "Newsletter channels" content source to create and activate new channels. Such a dataset is only used as a link to the UM's channel. Here the internal name of the channel in UM and the internal channel name in the dataset must be the same. If you create a dataset, a new channel is not created in UM. Equally, a new newsletter channel created in the project in UM must be manually added to the newsletter channels content source. The creating and management of the actual channels must continue to take place via UM. The channels maintained in the FirstSpirit project are merely for linking newsletter datasets and the channels in UM. This ensures that the generated event file refers to the right newsletter channel in UM.

Newsletter channels can be activated and deactivated in the FirstSpirit content source via the "Active" checkbox. Deactivated channels are ignored when the newsletter is sent.



ID	Name	Internal channel na...	Active
4352	Demo EN	newsletter_demo_en	☑
3712	Demo DE	newsletter_demo_en	☑

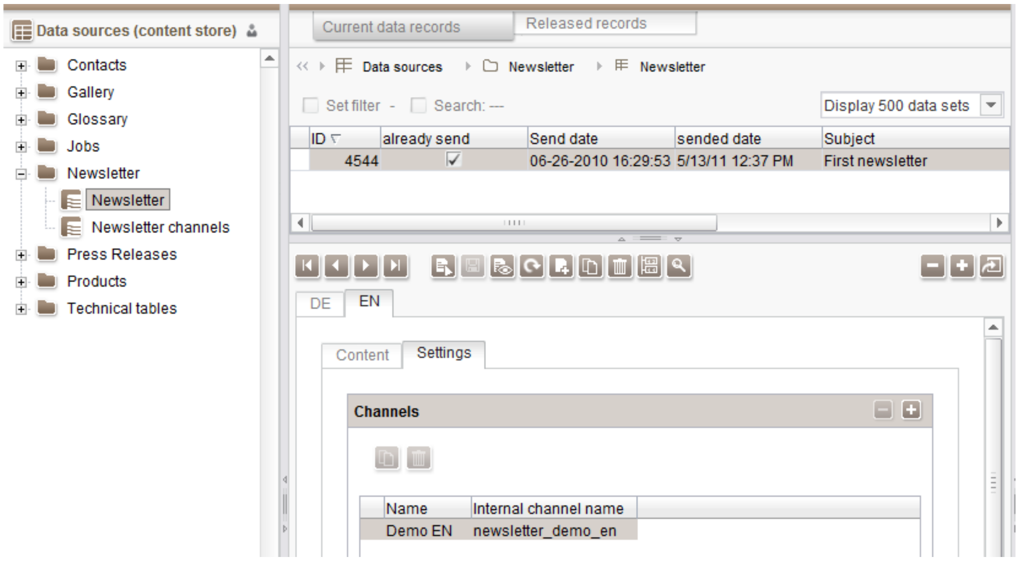
Name: Demo EN

Internal channel name: newsletter_demo_en

Active: ☑

Figure 7-1: The channels content in FirstSpirit JavaClient

A dataset for the newsletter must be created in the newsletter table. This must be linked to at least one channel, in order to define via which channel this newsletter is to be sent. To do this, the relevant channel must be added to the "Channels" in the content list. Apart from the optional fields such as "Sender" or "Reply to" (which are filled from the UM configuration, if they are empty), it is also possible to given the subject and content.



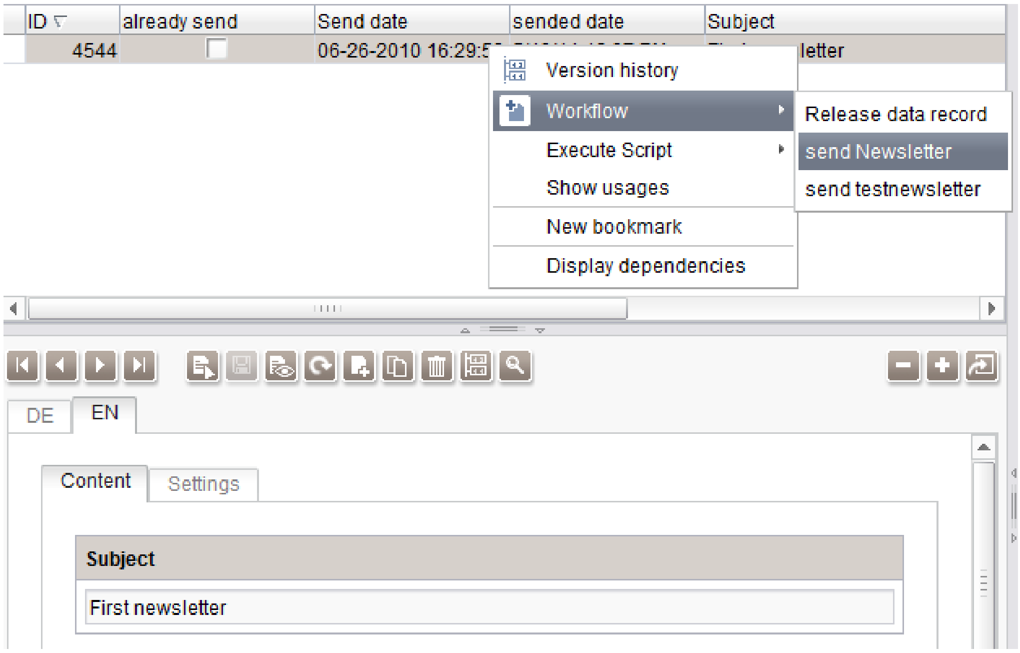
ID	already send	Send date	sended date	Subject
4544	✓	06-26-2010 16:29:53	5/13/11 12:37 PM	First newsletter

Name	Internal channel name
Demo EN	newsletter_demo_en

Figure 7-2: The newsletter content source

7.2 Dispatch of the newsletter

Before the newsletter can be sent, the relevant dataset must be released first. The dispatch is controlled via the workflow described in Chapter 5.3. This is run by right-clicking the newsletter dataset in the table.



ID	already send	Send date	sended date	Subject
4544	□	06-26-2010 16:29:5		letter

Subject

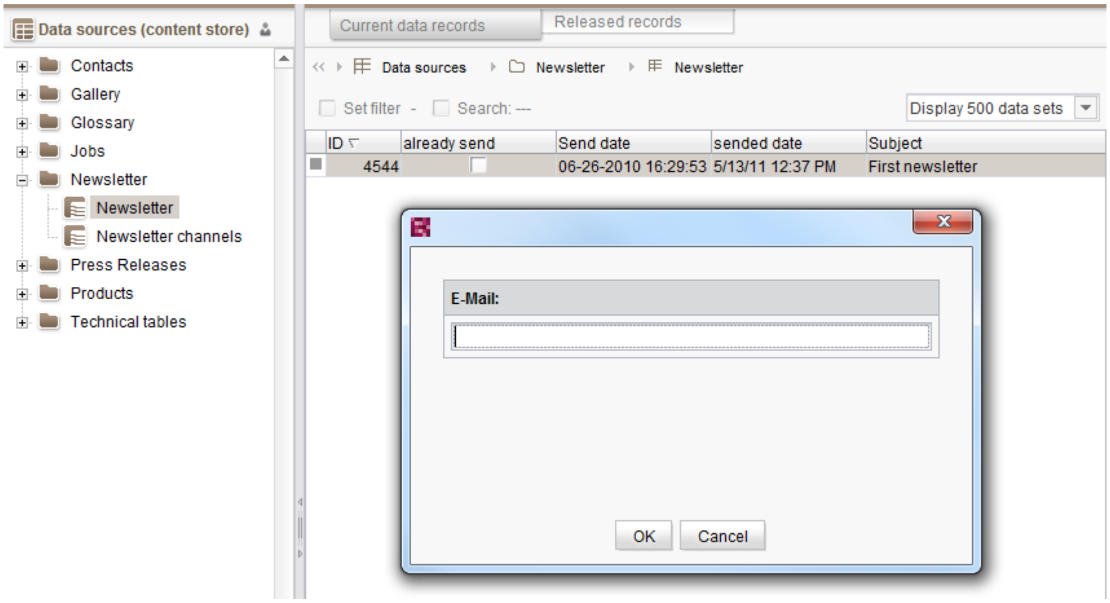
First newsletter

Figure 7-3: Starting workflow

If dispatch of the newsletter is approved with the help of the workflow, the workflow script is started, which triggers the generation of the selected dataset and its dispatch.

7.3 Dispatch of the test newsletter

In order to test a newsletter and its specific dispatch, it is possible to send a test newsletter to an address given by the user. To do this, the "Send test newsletter" workflow must be performed on the relevant newsletter dataset. Just like with the newsletter dispatch, the relevant dataset must be released first. Before the dispatch of the test newsletter is started, the user is prompted to give the destination e-mail address (see Figure 7-4)

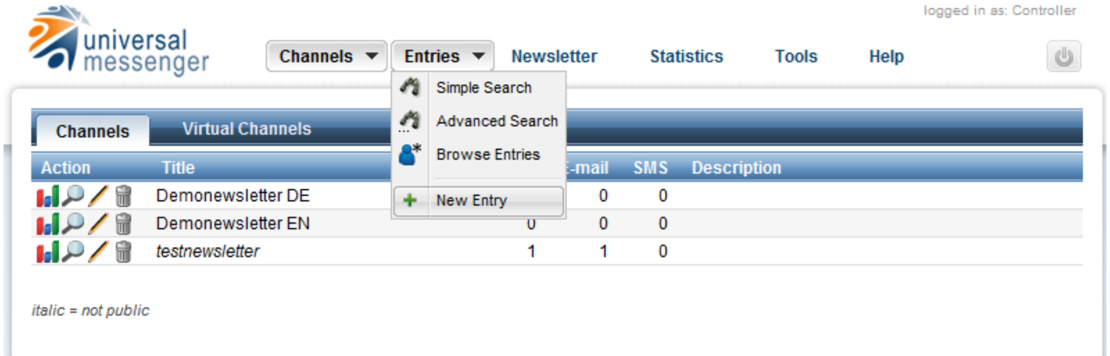


ID	already send	Send date	sended date	Subject
4544	☐	06-26-2010 16:29:53	5/13/11 12:37 PM	First newsletter

Figure 7-4: Sending the test newsletter

The destination address must be present as a mandatory entry in UM. The UM only sends newsletters to e-mail addresses present in the UM entries. This entry does not have to be added to the test channel. It is sufficient for the entry to be created.

A new entry can be created in UM under entries / New Entry.



Action	Title		-mail	SMS	Description
	Demonewsletter DE		0	0	
	Demonewsletter EN	0	0	0	
	testnewsletter	1	1	0	

italic = not public

Figure 7-5: Creating a new entry in UM

8 Legal notices

The "FirstSpirit™ EmailMarketing" module is a product belonging to e-Spirit AG, Dortmund, Germany.

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

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