webEdition manual

User Guide Standard

(c) living-e AG 2006

webEdition 4

User Guide

Standard 4.0.1
June 2006

© 2006 living-e AG
All rights reserved.

Printed in Germany

living-e AG does not extend any guarantee or warranty concerning the accuracy and correctness of any information contained in this manual.This manual may not be reproduced by any means, either electronically or mechanically, photocopied, recorded, reproduced or transferred by information storage or information retrieval systems by the purchaser, for purposes other than private use without the express written consent by living-e AG.The purchaser is authorized to make one backup copy of the software. A copy of this software can be requested from living-e AG at any time or it can be downloaded at www.webedition.de.Neither living-e AG nor suppliers of living-e AG shall be held liable for any damage (this shall include, without restriction, any damage arising from lost profits, operation breakdowns, loss of business information or data or any other financial losses), which arise from the use of this product or from the inability to use this product, even in the event that living-e AG has been informed of the possibility of such damage. In any case of damage, liability of living-e AG shall be limited to the amount you have actually paid for the product.

Microsoft Windows and Explorer are trademarks of Microsoft Corporation. Macintosh and Apple are trademarks of Apple Computer Inc. UNIX is a trademark of UNIX System Laboratories. All other trademarks are the property of their respective owners.

Contents

About this document 15

1 Introduction 19

What is webEdition? 19
How does webEdition work? 19
About webEdition database architecture and content storage 20
About webEdition templates and web page layout 20
About we:tags and content management 20
webEdition users: template administrators and Web page editors 21
The role of template administrators 21
The role of Web site editors 21
The workflow within webEdition 22
What's new in webEdition 4.0? 22
What's next... 23

2 User controls 25

Starting webEdition 25
Selecting your language 26
The webEdition main screen 26
Preferences 29
The User interface view 29
The File extension view 30
Editor view 30
Formmail recipients view 31
Proxy server view 31
Advanced view 31
The System tab 33
Error handling view 35
Backup view 36
Validation view 36
System settings for pageLogger 37
The webEdition menu structure 38
The File menu 38
The Options menu 41
The Modules menu 42
The Help menu 42
The explorer menu 43
Search function 45
Simple search 45
Advanced search 46
webEdition control elements 48
Viewing modes: standard and Super-Easy-Edit Mode 48
The editing screen in standard mode 48

The editing screen in Super-Easy-Edit Mode 49

3 Working with documents – the role of the editor 53

The four views when working with documents in normal mode 53
The Documents Properties view 54
The Documents Edit view 55
The Documents Information view 56
The Documents Preview view 57
The Documents Validation view 58
Working with webEdition pages 59
Creating a new webEdition document 59
Saving, “unpublishing”, and deleting documents 60
"Unpublishing" a file 61
Deleting documents, templates, objects, or classes 62
Overview of all unpublished pages 63
Modifying and updating a document 64
Creating a new document directory 64
Copying the contents of one directory to another 64
Rebuilding static and dynamic Web pages 65
Using the Import feature 65
Importing local files 66
Importing files from your server 68
Importing generic XML data 69
Importing CSV files 70
webEdition XML import 71
Working with graphical documents 73
Uploading an image 73
Importing more than one image 75
The image Properties view 77
The image Edit view 78
The images Information view 80
The images Thumbnails view: thumbnail formatting 80
Associating an image with a thumbnail format 82
Displaying your photo album 83
Working with non-graphical documents 84
Editing HTML pages 84
Using Flash movies 86
Using Quicktime movies 87
Editing JavaScripts 87
Creating a cascading style sheet (CSS) 88
Other files 89
Features in the Options menu 89
Document types 89
Categories 92
Changing user name and/or password 93
How an editor works within a template: using template input fields, lists and menus 93
Single-line input box (<we:input type="text">) 94
Date (<we:input type="date">) 94
Checkbox (<we:input type="checkbox">) 94
Input choice box (<we:input type="choice">) 94
Text box (<we:textarea>) 95
Lists (<we:block>) 102
Link list (<we:linklist>) 102
Select box (<we:select>) 105
Image field (<we:img>) 106
Movie fields (<we:flashmovie> and <we:quicktime>) 106

Editing documents in seeMode 106
The navigation tool 109
Using the navigation tool 109
The menus 110
Creating new navigation items 110
Creating a new navigation folder 114
Rules for the highlighting 117
Tags for the navigation tool 117
we:navigation 117
we:navigationEntry 118
we:navigationEntries 118
we:navigationField 119
we:navigationWrite 119
Example 119

4 Creating templates—the role of the administrator 123

What is a template? 123
What are we:tags? 123
How do I use we:tags? 123
The five views when working with templates 125
The template Properties view 125
The Templates Edit view 127
The Templates Information view 127
The Templates Preview Editmode view 128
The Templates Preview view 128
Creating Mastertemplates 129
Including templates 131
Template Rebuild 131
Creating templates 131
Creating templates using webEdition 132
Creating a new template 132
Saving a template 133
Updating/modifying a template 133
Deleting a template 133
Adopting an existing Web site 133
Integrating your own we:tags 135
Creating templates from documents made in HTML editing programs 135
Tags for the navigation tool 136
we:navigation 136
we:navigationEntry 136
we:navigationEntries 137
we:navigationField 137
we:navigationWrite 137
Example 137
Features for developing webEdition templates 139
Link lists (<we:link> and <we:linklist>) 139
Summary pages/areas 140
Creating a photo album 144
Producing and changing pages and objects from the front end 146
Including documents (<we:include>) 149
Search feature 150
E-Mailing a form 151
Working with HTML frames 152
Importing a Website 152
Generating templates automatically 153
Using a dedicated template 153

RSS feeds 155
Integrating external RSS feed 155
webEdition pages as RSS feed 156

Index 159

List of figures

Figure 1 The webEdition workflow 22
Figure 2 The webEdition login screen 25
Figure 3 Preferences: selecting your language from the User interface view 26
Figure 4 webEdition main screen showing menus, tabs and Cockpit screen 27
Figure 5 Cockpit start screen 28
Figure 6 File extensions view 30
Figure 7 The Editor view 30
Figure 8 The Formmail recipients view 31
Figure 9 The Proxy server view 31
Figure 10 Showinputs 32
Figure 11 Behaviour of the document type selection 32
Figure 12 The Advanced view 33
Figure 13 “Hint” text about the maximum upload size 33
Figure 14 pageLogger-directory in system tab 34
Figure 15 System tab 35
Figure 16 Error handling view 35
Figure 17 Backup view 36
Figure 18 Validation preferences 37
Figure 19 pageLogger-directory in system tab 38
Figure 20 The File menu 39
Figure 21 The File > New sub menu 39
Figure 22 The File > Open sub menu 39
Figure 23 The File > Delete sub menu 39
Figure 24 Import Wizard 40
Figure 25 The file manager 41
Figure 26 The Options menu 41
Figure 27 The Modules menu 42
Figure 28 The Help menu 42
Figure 29 Update window 43
Figure 30 The explorer menu 43
Figure 31 webEdition ID numbers 44
Figure 32 directory icon 44
Figure 33 webEdition page icon 44
Figure 34 Template icon 44
Figure 35 Image icon 44
Figure 36 movie icon 45
Figure 37 Plain text icon 45
Figure 38 Objects icon 45
Figure 39 Object class icon 45
Figure 40 Input field for search 45
Figure 41 The Search window - simple search 46
Figure 42 List of results 46
Figure 43 Activating advanced search 46
Figure 44 The Search window- advanced search 47
Figure 45 webEdition navigation control elements 48
Figure 46 The webEdition Edit view (normal mode) 49
Figure 47 What editors, administrators and visitors see 50
Figure 48 The No start document warning 51
Figure 49 The webEdition editing screen (seeMode) 51
Figure 50 Permissions for the seeMode in the User Management Module 52
Figure 51 The Documents Properties view 54
Figure 52 The Documents Edit view 56
Figure 53 The Documents Information view 57
Figure 54 The Documents Preview view 57
Figure 55 The Documents Validation view 58

Figure 56 Adding a new service 59
Figure 57 Creating a new document: Properties view 60
Figure 58 Deleting a document 62
Figure 59 Marking files for deletion 62
Figure 60 Quickstart-unpublished pages 63
Figure 61 Accessing unpublished pages from the File menu 63
Figure 62 Table of unpublished pages 64
Figure 63 Creating a document directory 64
Figure 64 Directory Properties view 64
Figure 65 The Import Wizard 65
Figure 66 Importing local files 66
Figure 67 Importieren Bildschirm 67
Figure 68 Dateien hinzufügen 67
Figure 69 List of selected files 68
Figure 70 Import successful 68
Figure 71 Importing files from a server 69
Figure 72 Generic XML Import 70
Figure 73 CSV Import 71
Figure 74 webEdition XML Import 71
Figure 75 webEdition XML import options 72
Figure 76 Import details 73
Figure 77 Uploading an image 73
Figure 78 Browsing your files for an image 74
Figure 79 Selecting an image to upload 74
Figure 80 Viewing an image in the Edit view 75
Figure 81 Import local files - step 1 of 2 75
Figure 82 Import local files - step 2 of 2 76
Figure 83 Import confirmation screen 76
Figure 84 The Properties view, Attributes area 77
Figure 85 The Properties view, Hyperlink area 78
Figure 86 Uploading a new image 79
Figure 87 Editing an image 79
Figure 88 Resizing an image 79
Figure 89 Rotating an image 80
Figure 90 Cropping an image 80
Figure 91 Converting and image 80
Figure 92 Defining thumbnail formats 81
Figure 93 Uploading a new image 82
Figure 94 Thumbnails tab 82
Figure 95 Formatting a thumbnail 83
Figure 96 Thumbnails list 83
Figure 97 Choosing a photo album directory 84
Figure 98 HTML Properties view 85
Figure 99 Editing an HTML page 85
Figure 100 Flash movie Edit view 87
Figure 101 Creating a CSS 88
Figure 102 Cascading style sheet in the Edit view 88
Figure 103 Creating a new document type 90
Figure 104 Naming a new document type 91
Figure 105 New document type/delete document type 91
Figure 106 Selecting a template for a document 91
Figure 107 The Categories feature 92
Figure 108 Change user name screen 93
Figure 109 Change password screen 93
Figure 110 Sample page of single-line input box in the Edit view 94
Figure 111 Sample page showing date fields 94

Figure 112 Using an input choice box 95
Figure 113 Using a text box 95
Figure 114 Using a WYSIWYG input screen 96
Figure 115 WYSIWYG toggle buttons 96
Figure 116 WYSIWYG textfield 96
Figure 117 Define abbreviation for a screenreader 97
Figure 118 Language selection for screenreader 97
Figure 119 Edit Hyperlink in WYSIWYG 98
Figure 120 Insert table - table summary 99
Figure 121 Edit table cell dialogue box 100
Figure 122 Table caption 100
Figure 123 Import .rtf file button 100
Figure 124 The Import RTF file dialogue box 101
Figure 125 Example of an RTF file 101
Figure 126 RTF text in Preview mode 101
Figure 127 RTF text imported into webEdition 102
Figure 128 Example of a list block 102
Figure 129 Example of a list block in Preview view 102
Figure 130 Example of a link list (we_demo/includes/navigation.php) 103
Figure 131 Example of a link list in Preview view 103
Figure 132 Edit hyperlink button 103
Figure 133 Edit link dialogue box 104
Figure 134 Edit Link 105
Figure 135 Linking images in version 3.2 105
Figure 136 Select box 105
Figure 137 The image field 106
Figure 138 The Flash and Quicktime movie fields 106
Figure 139 The seeMode preview 107
Figure 140 The seeMode included file list 107
Figure 141 Login to seeMode 108
Figure 142 seeMode edit view 108
Figure 143 Menu navigation tool 109
Figure 144 The naviagtion tool 109
Figure 145 Navigation in the navigation tool and on the website 110
Figure 146 View Properties for new entry 111
Figure 147 Select navigation folder 111
Figure 148 Selection navigation entry 112
Figure 149 Example entry 112
Figure 150 Subnaviagtion item 112
Figure 151 Area Selection - Dynamic selection 113
Figure 152 Preview of the navigation entries 114
Figure 153 Navigation on the website 114
Figure 154 New folder 114
Figure 155 Dynamic contents in folder 115
Figure 156 Dynamic navigation folder in the explorer tree 116
Figure 157 The Preview view 116
Figure 158 Dropdown Rules for highlighting 117
Figure 159 Window Navigation Rules 117
Figure 160 Navigation we_demo 121
Figure 161 The Tag Wizard 124
Figure 162 Assigning attributes using the Tag Wizard 125
Figure 163 The Templates Properties view 126
Figure 164 Copying a template using the file manager 126
Figure 165 The Templates Edit view 127
Figure 166 The Templates Information view 127
Figure 167 The Templates Preview Editmode view 128

Figure 168 Template Preview view 129
Figure 169 Charting 130
Figure 170 Properties of a template 131
Figure 171 Template Rebuild 131
Figure 172 Example of output for a link list 140
Figure 173 Sample summary page 142
Figure 174 Sample of all summary pages 143
Figure 175 Sample input box 150
Figure 176 The Formmail recipients view 151
Figure 177 Import Wizard 152
Figure 178 Checkbox bei Import 152
Figure 179 Automatically generating templates 153
Figure 180 Specified template 154

List of procedures

Procedure 1 Creating a new document 59
Procedure 2 Importing local files 66
Procedure 3 Importing files from your server 68
Procedure 4 Importing generic XML data 69
Procedure 5 Importing CSV files 70
Procedure 6 webEdition XML import 71
Procedure 7 Adding and Editing links 103
Procedure 8 Editing a document using seeMode 107
Procedure 9 Creating a new template 132
Procedure 10 Adopting an existing page 134
Procedure 11 Create your own we:tag 135
Procedure 12 Creating a template from an HTML editor 135

About this document

Purpose of this document

The webEdition User Guide provides an overview of webEdition's features and an introduction to all the processes you will use to operate the Standard Version of webEdition. This manual has been designed to include a complete description of how to use webEdition to create new Web sites, and how to adopt and modify existing Web sites.

This User Guide has the following sections:

- Introduction to webEdition
- User controls and description of the webEdition interface
- Role of the webEdition editor
- Role of the webEdition administrator
- List of important terms

For additional information about the installation process and the suite of webEdition modules, see the *webEdition Installation, Update and Backup Procedures*. This document is available on the webEdition documentation Web page.

Audience

This document is intended for personnel in the following groups:

- Web administrators
- Web editors

The webEdition customer documentation suite

The documentation team publishes new webEdition documents to support the release of all webEdition features, modules and enhancements.

You can consult our documentation suite for detailed information about the modules you have purchased or about webEdition products that you may wish to purchase in the future. All customer documentation is available in portable document format (PDF) on the webEdition documentation Web page.

On-line reference documentation

The webEdition customer documentation suite comprises the following books, all of which you can obtain at URL:
http://www.webedition.de/english/downloads/Documentation.html

Standard webEdition documentation

The following books support the webEdition Standard suite:

- *The webEdition User Guide*
- *The webEdition Installation, Update and Backup Procedures*
- *The webEdition Tag Reference*

Documentation for webEdition modules

The following books support the webEdition modules:

- *The Customer Management and Customer Management PRO Module User Guide*
- *The Database/Object Module User Guide*
- *The Editor Plug-in User Guide*
- *The Newsletter Module User Guide*
- *The Scheduler and Scheduler PRO Module User Guide*
- *The Shop Module User Guide*
- *The Task/Messaging Module User Guide*
- *The User Management and User Management PRO Module User Guide*
- *The Export Module User Guide*
- *The Voting Module User Guide*
- *The Workflow Module User Guide*

What precautionary messages mean

webEdition documents include attention and caution messages, which are designed to draw your attention to important instructions.

Attention boxes

An attention box identifies information that is necessary for the proper performance of a specified task. For example:

> **ATTENTION**
>
> You must have the appropriate permissions in your user profile to complete this procedure. Permissions are assigned to you by your webEdition system administrator. Contact your webEdition system administrator for further details.

Caution boxes

Caution messages indicate that there are possible risks to your software or database if you perform a specified task without taking the suggested appropriate precautions. For example:

CAUTION
Database warning
If you complete this procedure, your database will be overwritten.

How commands, system responses and we:tags are represented

The commands, system responses and webEdition tags (called we:tags) used in this document conform to the following conventions.

Web interface commands

Commands performed through a Web browser are shown in *italic* typeface. For example:

Click on the *Save* button.

Menu-based commands performed in a Web browser are shown in *italic* typeface. Drop-down or nested menu commands are separated by an input prompt (>). For example:

Select *Customers* > *New* from the main menu of the Customer Management Module.

webEdition tags and template code

The webEdition templates use a specialized set of programming tags based on the PHP programming language. These webEdition tags or we:tags are displayed in `courier` typeface and in angled brackets:

Opening tags appear thus: `<we:tag/>`

Closing tags appear thus: `</we:tag>`

The programming code used in webEdition templates is also represented in this document by `courier` typeface:

```
<we:sessionStart/>
<we:ifRegisteredUser>
Hello: <we:sessionField Last name="user name" type="print"/><br>
Logged in
</we:ifRegisteredUser>
```

Attribute variables

Attributes and variables appear in *`courier italic`* typeface. For example:

`<we:hidden name="`*`attribute1`*`">`

How to check the document version and issue

The information on the title page of this document indicates the version and issue for this publication. The version and issue also appear in the footer on every even-numbered page.

The first two digits in the document numbering scheme indicate the version. The version number increases each time the document is updated to support a new software release. For example, the first release of a document is 01.01. In the next software release cycle, the first release of the same document is 02.01.

The second two digits in the document numbering scheme indicate the issue. The issue number increases each time the document is revised and re-released in the same software release cycle. For example, the second release of a document in the same software release cycle is 01.02.

Customer service

For further information about webEdition, please consult our Web page, or contact our customer service department:

- Web Page:http://www.webedition.biz/
- E-mail:
  - Technical support: technik@webEdition.de
  - Sales: sales@webEdition.de
  - Information/Help: info@webEdition.de

1 Introduction

This introduction is intended to help new users familiarize themselves with the webEdition content management software. If you have used previous versions of webEdition and are already familiar with the software, you can skip to Section 1.4, "What's new in webEdition 4.0?" on page 22.

This chapter treats the following topics:

- Section 1.1, "What is webEdition?" on page 19
- Section 1.2, "How does webEdition work?" on page 19
- Section 1.3, "webEdition users: template administrators and Web page editors" on page 21
- Section 1.4, "What's new in webEdition 4.0?" on page 22
- Section 1.5, "What's next..." on page 23

1.1 What is webEdition?

webEdition is a database-grounded Web content management system (CMS). A CMS is a software application that allows you to develop, administer and control the information and the format of your Web page. Unlike file-based systems, which write new content to new HTML pages and then manipulate the content using a complex management system, webEdition stores all content information (such as text, images, graphics, Flash movies, style sheets) in a central database.

The webEdition program only has to be installed once on the server of your service provider. This is done using a simple installation program run on a computer with Internet access. All operations thereafter are performed through a standard Web browser. There is no need for you to install any additional software on your local terminal.

The architecture of webEdition reflects the philosophy of separating content and layout. Your web content (text, images, etc.) is stored in a database; the layout of your web pages is controlled by templates.

1.2 How does webEdition work?

The following subsections describe how the webEdition CMS operates from the point of view of content architecture, layout and management. These concepts are integral to understanding how webEdition works.

1.2.1 About webEdition database architecture and content storage

webEdition uses MySQL, a database system that is supported by almost all major Internet service providers. There are several advantages to this architecture. Because all of your content is stored in a database, the HTML-coded web page itself does not have to be edited if the content has to be changed. An editor only has to change the database entry using the simple WYSIWYG webEdition interface and then re-save the corresponding page. In addition, the content stored in the database can be rapidly searched (and modified) or used in other ways (for example, as a central data repository). Moreover, the database runs primarily in the background, and once installed, its operation is transparent to the front-end user.

1.2.2 About webEdition templates and web page layout

webEdition works with templates. Templates are pages programmed using HTML and PHP that act as a guide to form the structure of a Web page. These pages contain the basic layout for your Web pages. Templates impose a consistent format upon your Web site, and permit only certain areas of the page to be edited.

Templates are normally created using HTML and expanded using we:tags. Non-variable areas of your page are coded in HTML. The areas you wish to edit or change (such as text, images, graphics) are marked with we:tags. The we:tags will act as placeholders for variable data.

If you are setting up a new web site, you can either create the templates yourself or you can have them created by your internal design department or by an agency. You can set up your entire Web site in webEdition, since it also acts as a completely functional HTML editing program. Templates developed using other HTML editing programs can be imported directly or merely copied and pasted to webEdition. Importing an existing Web site is also possible by merely inserting the appropriate we:tags where the hard-coded text, images, link lists etc. are located.

1.2.2.1 About graphics and images

webEdition differentiate between layout images and images/graphics that are dependent on content. Layout images have no connection to the content and only affect the appearance of the page. Layout images that are already present in the templates do not have to be imported to webEdition, since they do not have to be managed using the database. If, however, something in the layout needs to be modified, the template must be changed and re-saved.

Unlike layout images, content-related images or graphics (such as product photos or charts) are managed and edited with the webEdition system in the same way as texts.

1.2.3 About we:tags and content management

Web site content is managed using webEdition tags (we:tags). They comprise a syntax based on XML that stores functions in an actions library and then integrates them when the page is called. This system makes creating text areas, summary pages, or images a matter of entering a few we:tags. The we:tags are very simple in structure and are easily understood by anyone with a basic understanding of HTML. To make finding, understanding and using we:tags easy, webEdition also provides a Tag Wizard along with an on-line *Tag Reference Guide*.

The we:tags are only used by the administrator when a template is being created. Thereafter, the editor who will work with the templates to enter or edit data for the web pages does not have to be familiar with the we:tags.

1.3 webEdition users: template administrators and Web page editors

webEdition differentiates between the role of the editor and that of the administrator. Each role requires a different skill-set, and the tasks assigned to each role are governed by different webEdition user permissions. Depending on the size of your company or organization, the editor and the administrator can be one and the same person.

1.3.1 The role of template administrators

The template administrator is responsible for the design and creation of the templates. He or she is required to have a basic understanding of HTML, since the webEdition tags (we:tags) have to be inserted in a pages created using HTML.

Administrators also perform the following system-level tasks:

- Installing modules
- setting user permissions
- performing backup and restore functions

Note: The first person to install webEdition automatically has administrative rights for the system. For details regarding installation procedures and system-level permissions, see the *webEdition Installation, Update and Backup Procedures*. For an overview of the user-level controls available in the extended webEdition suite, see the *User Management Module User Guide*.

1.3.2 The role of Web site editors

Editors are generally involved in creating and updating Web documents. The editor only has to know what the input boxes look like on the webEdition templates in order to fill the fields with text. Their tasks can also include such things as integrating images or generating link lists, but these tasks do not require any programming knowledge. The webEdition editors need not concern themselves with the database; they work directly with their Web sites using the input boxes on their actual Web pages.

If editors want to create a new page, they simply call up the template to be used for the page (so they do not have to worry about the layout), enter the desired text and/or images, and then save the new page. This process does not require editors to be familiar with HTML editing programs, or FTP.

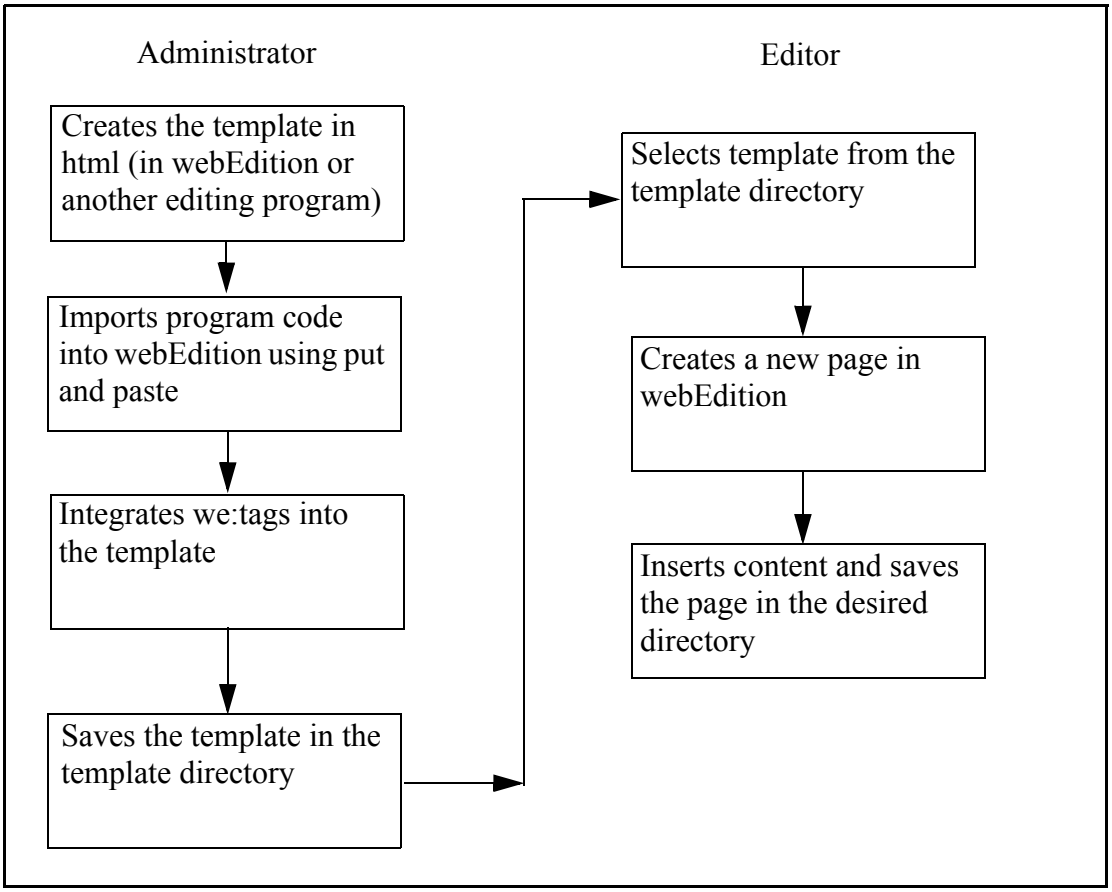
Note: If necessary, a differentiation may be meaningful between administrator and developer. In this case the administrator would be the contact for the technical interests of your web site and thus responsibly for installation, maintenance and module installation of webEdition. Consequently it would be the administrator who would assign the user's rights within webEdition. The developer would be responsible only for the production and maintenance of the templates. In practice the borders are fluent between these both functions, especially in smaller companies. Thus it is not unusual that in such a constellation the developer also installs webEdition and the user's rights for the editors. For simplification the functions are

summarized by administrator and developer in the present documentation under the upper concept Administrator.

1.3.3 The workflow within webEdition

A “workflow” refers to the activities a page must to pass through before it can be published. Figure 1 shows a diagram portraying a simplified workflow with webEdition and the roles played by both the administrator and the editor.

Figure 1 The webEdition workflow



1.4 What’s new in webEdition 4.0?

If you have upgraded to webEdition 4.0, you will find the following changes and enhancements to the user interface:

- *Internal search function*: Particularly with bigger projects it is sometimes difficult to preserve the overview about many documents, templates, objects and classes. Hence, the new search function is a considerable working relief. You find closer information in .
- *The cockpit*: The improved start screen in webEdition 4 is the cockpit. From the first sight you have immediately access to many important functions. In addition you find closer information in .
- *Navigation too*l: Simplifies production and maintenance of the navigation of your web page . You find more information in .

- *Improved Tag Wizard*: The Tag Wizard was completely reworked for webEdition 4 to present the information more clear. You find further details to the Tag-Wizard in
- *Master template*: Keep your design more consistent by using the new Master Template feature

1.5 What's next...

The rest of this *Guide* is written with a user-based workflow model in mind. The next chapter provides a description of the user interface. Thereafter, one chapter has been written for editors, who will be working with the finished templates and who do not need to be able to use we:tags. The other chapter is intended for administrators, who create the templates and are responsible for ensuring the integrity of their Web sites.

2 User controls

This chapter describes the webEdition user interface and how to use its basic controls. It is intended for both editors and template administrators. This chapter treats the following topics:

- Section 2.1, "Starting webEdition" on page 25
- Section 2.2, "Preferences" on page 29
- Section 2.3, "The webEdition menu structure" on page 38
- Section 2.6, "Viewing modes: standard and Super-Easy-Edit Mode" on page 48

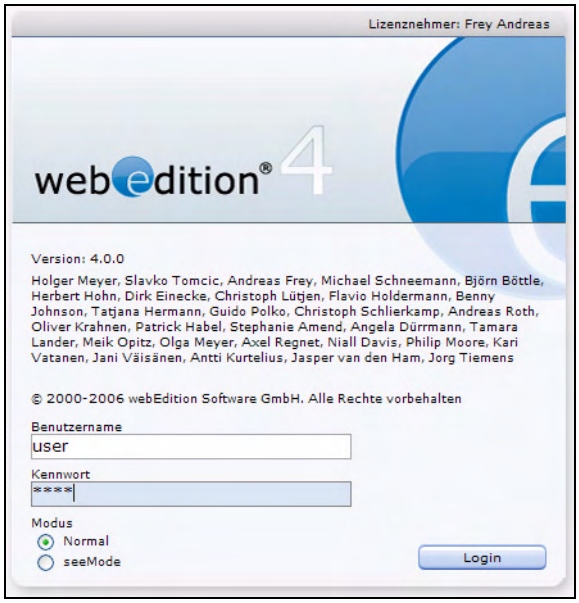
For information on installing webEdition, see *The webEdition Installation, Update and Backup Procedures*.

Specific information about how to create and maintain webEdition documents is found in Chapter 3, "Working with documents – the role of the editor" on page 53. Information on how to create and use templates is found in Chapter 4, "Creating templates—the role of the administrator" on page 123.

2.1 Starting webEdition

In your browser enter www.yourdomain.com/webEdition/. The login screen appears (see Figure 2). Enter your User name and Password in the designated fields.

Figure 2 The webEdition login screen



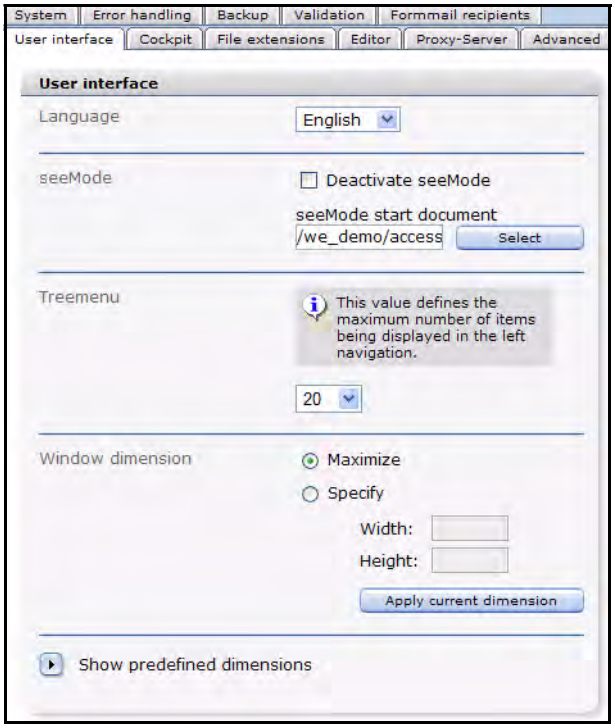
The first time you enter your user name and password, you always enter "admin" and "admin". After entering you user name and password, click *Login*.

At the bottom of the screen under the word *Mode* there are two radio buttons where you are given the option of choosing the viewing mode you wish to use, either *Normal* or *seeMode*. For more information about the two viewing modes in webEdition, see Section 2.6, "Viewing modes: standard and Super-Easy-Edit Mode" on page 48.

2.1.1 Selecting your language

Choose *Options* > *Preferences*.... The *Preferences* window opens. Click on the *User Interface* tab. The User Interface view appears (see Figure 3).

Figure 3 Preferences: selecting your language from the *User interface* view



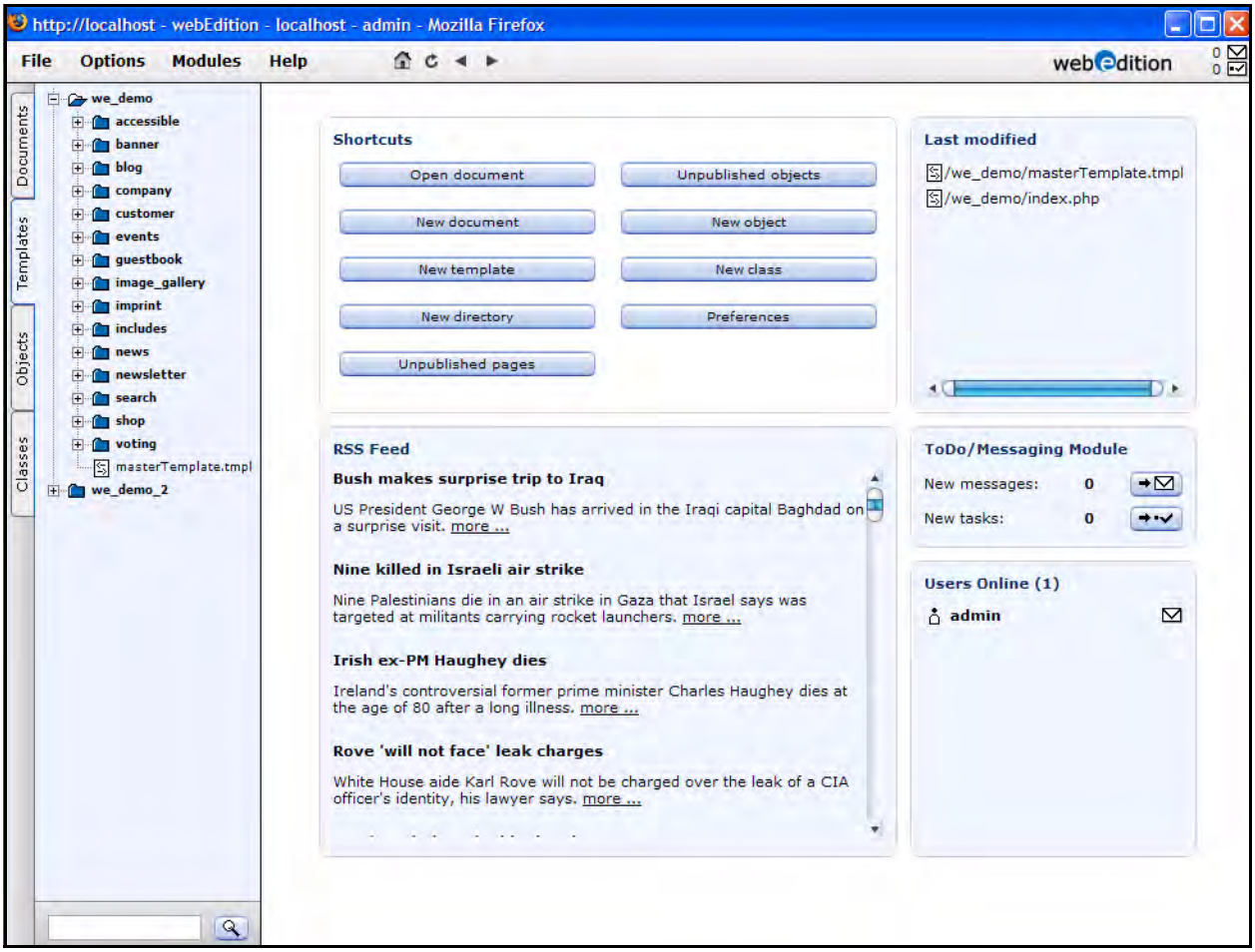
In the *Languages* area, select your language from the select box. Click the *Save* button at the bottom of the window. After saving your language preference, the webEdition display changes to the language you have selected.

Note: Information about other functions in the *Preferences* window is presented in Section 2.2.

2.1.2 The webEdition main screen

The controls for webEdition can be found on the webEdition main screen in the drop-down menus *File*, *Options*, *Modules* and *Help*, located at the top of your browser window (see Figure 4). To the right of the menus, are the navigational control icons *Home*, *Reload*, *Back* and *Forward*. Below the drop-down menus on the left side of the screen, is the explorer menu which displays the structure of your Web site. To the right of the explorer menu is the editing screen (which is explained in Section 2.6.1, "The editing screen in standard mode" on page 48).

Figure 4 webEdition main screen showing menus, tabs and *Cockpit* screen



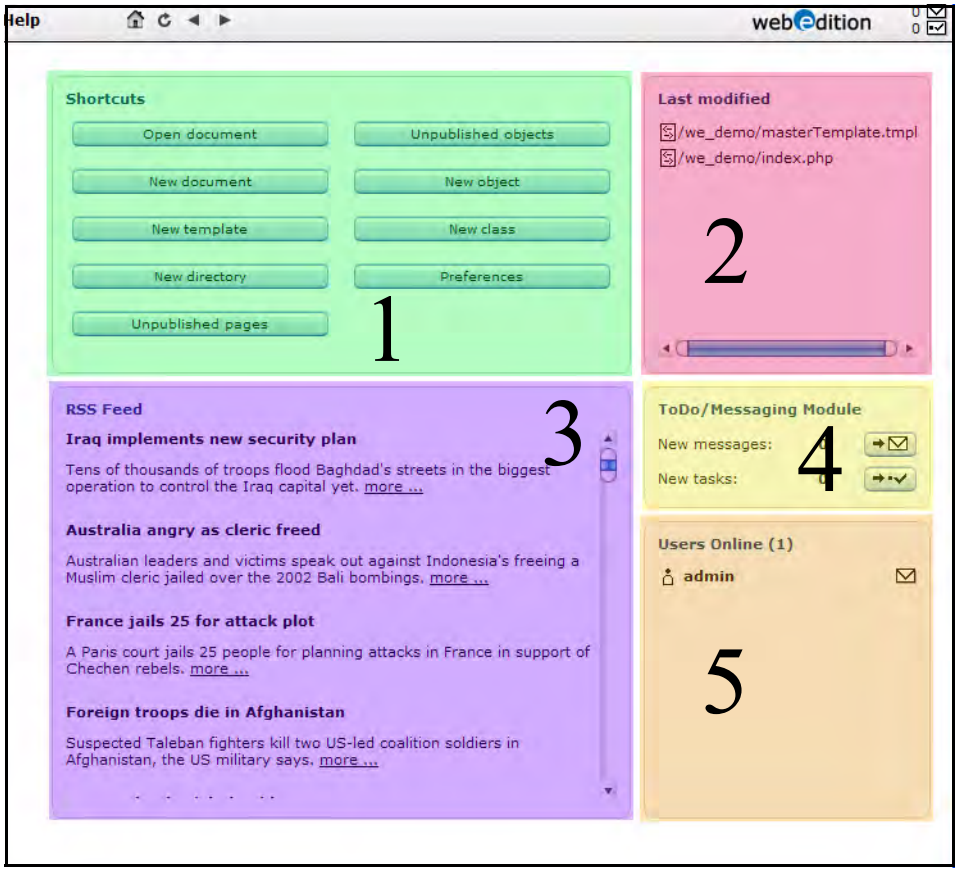
Note 1: For your browser to be able to display the menu bar, it must be Java-compliant. If you do not have Java installed on your browser and do not plan to install it, the webEdition menus will be displayed as three individual drop-down menus. They will appear in the same order, but will only have one level.

Note 2: If you have only just installed webEdition, the menu will only display the we_demo directories with our sample Web site. All examples in this manual are based on we_demo. Your own pages will be set up in the root directory and not in the we_demo directory.

2.1.2.1 The *Cockpit*

From Version 4 of webEdition, a revised start display - the cockpit - is available to you which summarizes with every login to webEdition the most important information on a glance:

Figure 5 Cockpit start screen



- (1) *Shortcuts*: These options are dispalyed right after logging in. There are the following actions:
  - — *Open document*. Opens a document or template
  - — *New document*. Creates a new document
  - — *New template*. Creates a new template
  - — *New directory*. Creates a new directory
  - — *Unpublished pages*. Shows an overview of unpublished pages
  - — *Preferences*. Opens the Preferences dialogue box

  The following items are only displayed, if the DB/Object Module is installed:
  - — *New Object*. Creates a new object
  - — *New Class*. Creates a new class
- (2) *Last modified*: Here, the latest edited documents, templates, object and classes are displayed. You can determine the number of entries displayed in the *Preferences*. A click on an element leads directly to the corresponding entry.
- (3) *RSS Feed*: Here, any selected RSS feed can be displayed.

- (4) *ToDo/Messaging Module*: Your newly received ToDos or messages are indicated here (the ToDo/Messaging Module must be installed). Click on the suitable button to write immediately your own messages or ToDos
- (5) *Users Online*: Shows the currently registered users in the system. With a click on the icon beside a user you can write a quick message to that user

Note: The numbers in brackets relate to the elements in Figure 5.

You may return to the webEdition Cockpit by selecting *File* > *Cockpit*.

2.2 Preferences

Select *Options* > *Preferences...* to open the *Preferences* screen (see Figure 3, "Preferences: selecting your language from the User interface view" on page 26).

Note: Some preferences are restricted to administrators. The preferences that are visible are dependent upon the User Management Module that you have installed.

- If you have installed either the User Management or the User Management PRO Module, the following preferences are available only to the administrator: *Proxy*, *Advanced*, *Modules*.
- If you have installed the User Management Module, the following preferences are available only to the administrator: *seeMode*, *File Extensions*, *Formmail recipients*.
- If you have installed the User Management PRO Module, the following preferences can be set by users if special permissions have been granted by the administrator: *seeMode*, *File Extensions*, *Formmail recipients*.

The *Preferences* screen has the several tabs (see Figure 3, "Preferences: selecting your language from the User interface view" on page 26); the functions of which are described in the following subsections.

2.2.1 The *User interface* view

The *User interface* view has the following areas (see Figure 3, "Preferences: selecting your language from the User interface view" on page 26):

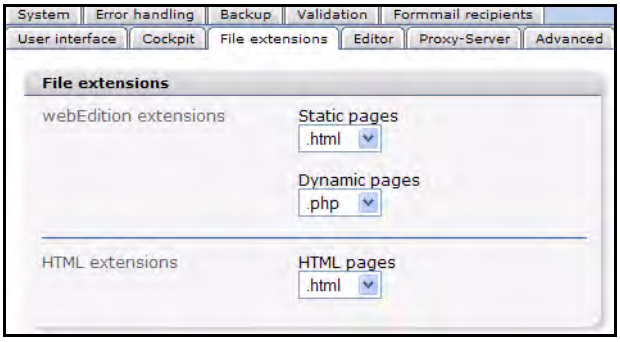
- *Language*. Use this select box to choose the language of the user interface.
- *seeMode start document*. When you login into seeMode, webedition needs to know what document it should display at startup. If no start document is set and you login into seeMode, webEdition automatically asks you for the start document.

  You can deactivate the seeMode by clicking on the *Deactivate seeMode* checkbox. If you do so, the seeMode option will not be available to you on the login screen
- *Treemenu*. Define the maximum number of items being displayed in the left navigation.
- *Window dimension*/*Predefined dimensions*. These settings allow you to manipulate the appearance of the webEdition interface. Select the *Maximize* radio button and the main webEdition window always opens as large as possible. If you prefer to have the window open in a different size, select *Specify* and set the pixel values or use the *Predefined dimensions* area to choose from commonly used pixel values.

2.2.2 The *File extension* view

The *File extensions* view has the following areas (see Figure 6):

Figure 6 *File extensions* view

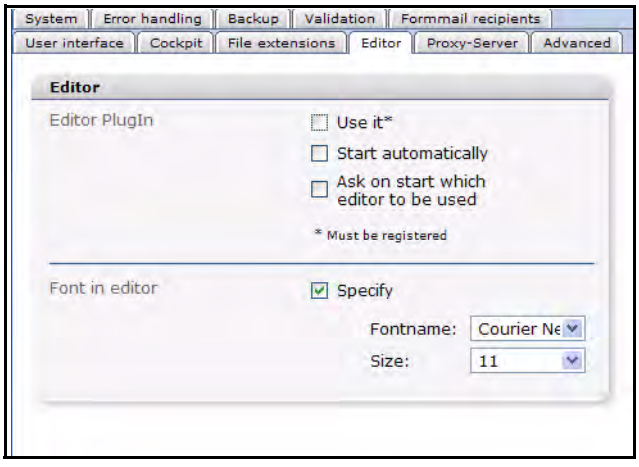


- *webEdition extensions*. In this section, you can specify the default extension for static and dynamic pages.
  - — *Static pages*. Choose the default extension for static pages from the select box. This value is generally .html or .htm for HTML pages. You can also set .asp for Active Server Pages or .jsp for Java Server Pages, if necessary. However, this should only be done if the appropriate code has been implemented in the template. If you wish to use PHP code in your template, you will have to store the page as a dynamic page (see the next item).
  - — *Dynamic pages*. Specify the extension for dynamic pages (generally .php or .PHP4). This extension will cause the Web server to execute PHP4.
- HTML extensions: Specify the default setting for HTML pages. If you primarily use PHP scripts or other similar scripts, you can also select the corresponding extension.

2.2.3 *Editor* view

The *Editor* view has the following areas:

Figure 7 The *Editor* view



- *Editor-Plug-in*. This feature is only available if you use Microsoft Windows with the Internet Explorer, and if you have purchased a licence from webEdition to use the Plug-in. Activate the *Start automatically* checkbox to load the Plug-in every

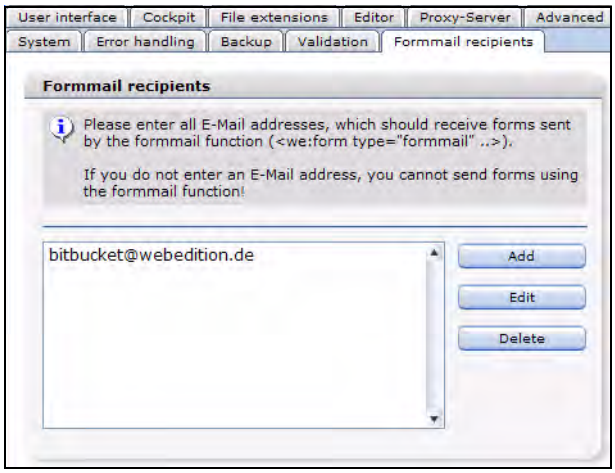
time you click the *Edit* tab in a template. Activate the *Ask on start...* checkbox, and webEdition will ask you which editor to use every time the Plug-in is started.

- *Editor dimensions*/*Predefined dimensions*. These controls operate like the *windows dimensions* described under the *User interface* view.

2.2.4 *Formmail recipients* view

With the *Formmail recipients* view, you can enter the addresses to which formmail can be sent. If no addresses are specified, the formmail function is blocked. If you have used the formmail function in prior versions of webEdition, you must enter the addresses which are defined in the template (<we:form type="formmail" ....> under "recipients") in the formmail recipients function. For further details, see Section 4.6.7, "E-Mailing a form" on page 151.

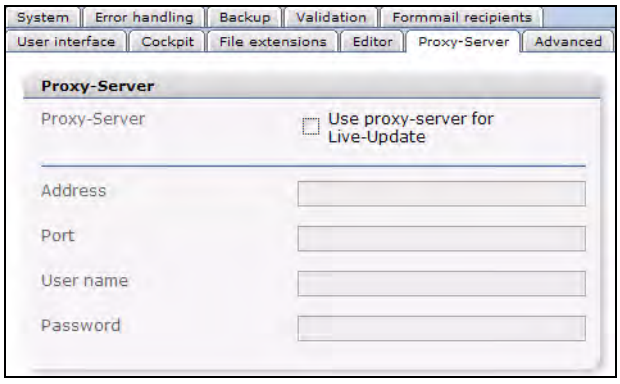
Figure 8 The *Formmail recipients* view



2.2.5 *Proxy server* view

Set the *Proxy server* view preferences if your Web server requires a proxy server to create a connection to the Internet (as in the case of a corporate network, or intranet).

Figure 9 The *Proxy server* view



2.2.6 *Advanced* view

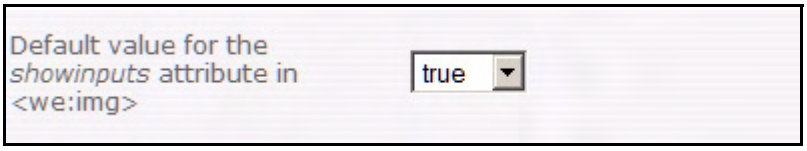
The *Advanced* view has the following areas:

- *Default settings for the PHP attribute in we:tags*. You can assign whether or not the editor ought to be given the right to enter PHP code in an entry field (for example,

<we:input type="text/>, <we:textarea/>). Generally the setting (default = false) is such that the editor is prohibited from writing PHP code in the entry field. In order to override this restriction, set the value in the drop-down menu to "true". It is also possible to override this restriction by using the we:tags (for example: <we:input type="text" php="true"/>).

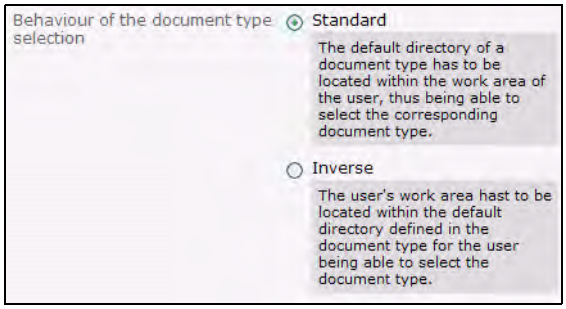
- *Default value for the inlineedit attribute in <we:textarea>*. If you select *true*, you can edit the textarea directly on the same Web page. If you select *false*, you have to click first on the *Edit* button. A new window will open where you can edit the textarea. If there are many textareas, the value *false* is recommended, as the web page will load much faster.
- *Use Safari Wysiwyg editor (beta Version)*. Set this option to *true* to activate the Safari WYSIWYG editor. This Function is still in beta state.
- *Default value for the showinputs attribute in <we:img>*. Set this attribute to *true* to display to your web site editors the *Alt* (alternative text) and *Title* fields for images if they are wrking with accessible or barrier-free sites. The *show inputs* field can be toggled to false (off), if you are not working with an accessible web site and want to keep unneccessary fields from the view of your editors.

Figure 10 Showinputs



- *Behaviour of the document type selection*. Use this field to define the document types available to your users.

Figure 11 Behaviour of the document type selection



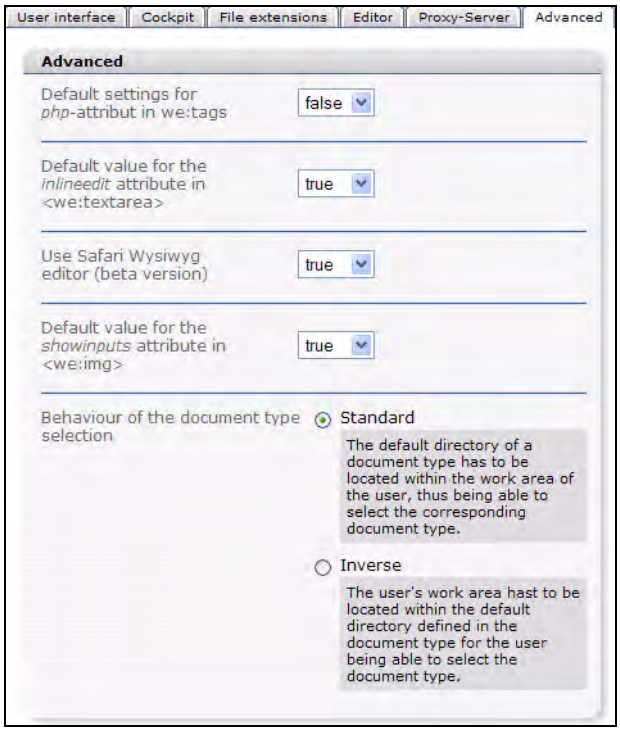
- *Type of database connections*. Select the method in which you will connect to the MySQL database. There are two types:
  - — *connect*. Normal connection to the database. The system will connect to the database using the server address, MySQL account name and password that has been entered into webEdition by the administrator when the program was installed. If the connection fails, the program will stop running.
  - — *pconnect*. Persistent database connection. This type of connection behaves similarly to normal connections with two exceptions. Before a new connection is made to the database, the system attempts to use an existing persistent connection to the same database with the same user name and password. If no such idle connection exists, a new connection is established. Second, the

connection to your MySQL server is not closed when the PHP script exits. It remains idle to be used at a later stage.

Note: If you are unsure about what type of connection to use, choose *connect*.

- *HTTP authentication*. Activate the checkbox if your server uses HTTP authentication. Thereafter, enter your username and password in the required fields.

Figure 12 The *Advanced* view



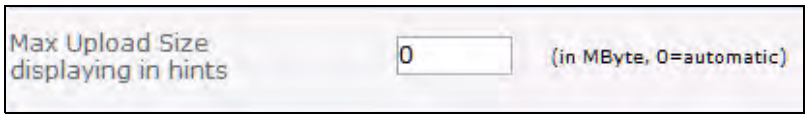
2.2.7 The *System* tab

The System view has the follwing items.

-

- *Max Upload Size displayed in "hints"* (see Figure 13 on page 33). In order to ensure that when uploading files to their Web site, editors do not exceed the maximum upload limits set for your PHP server, webEdition allows the administrator to enter a value that will be displayed in the hint (or mouse-over) text as guidance for editors.

Figure 13 "Hint" text about the maximum upload size

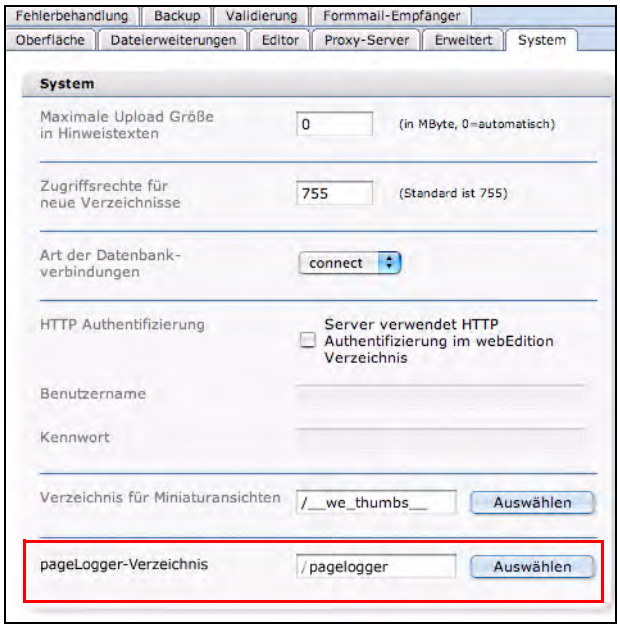


Note: The values provided by the server and PHP are not changed here; here the administrator defines a maximaum value (that is, a value less than that prescribed by the server) which is intended to help the editors.

- *Thumbnail directory*. Select a directory for thumbnails.
- *pageLogger directory*.

  The *pageLogger* area has been added to the *System* tab: *pageLogger-directory* (see Figure 19). Please select here the directory, where pageLogger is installed.

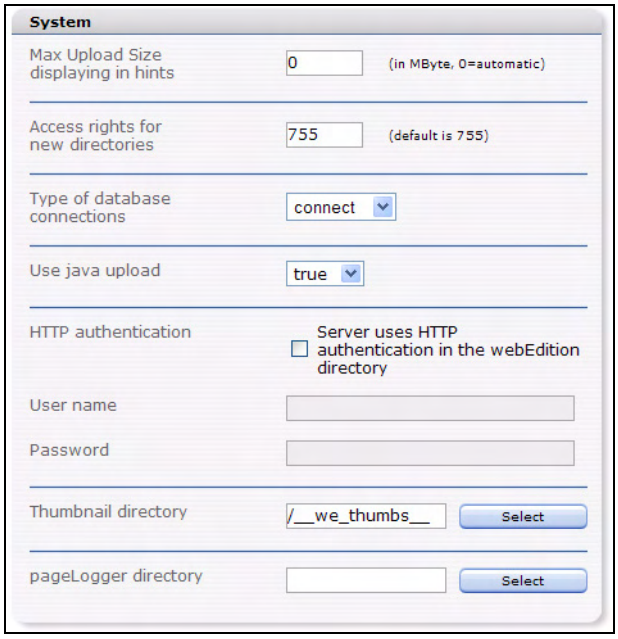
Figure 14 pageLogger-directory in system tab



As soon as the pageLogger directory is set here correctly, you are able to start pageLogger simply by clicking on *Modules* > *pageLogger*. Pleas note, that you have to restart webEdition for the changes to take effect.

> ***Note:*** For the *Single Login* feature to work correctly, a pageLogger user with the same name as the webEdition user has to be established. Furthermore, the user has to be marked as a *webEdition user* in the user preferences of *pageLogger*! For further information about webEdition users in pageLogger please consult the pageLogger documentation.

Figure 15 *System* tab



The areas *Max Upload Size displaying in hints*, *Type of database connections*, *HTTP authentication* and *Thumbnail directory* are taken off the *Advanced* tab to improve the clarity. You can now set the *Acess rights for new directories*. The default value is *755*. For more information about access permissions in Unix file systems, please ask your system administrator.

Note: The permissions set in the *Systems* view only apply to newly created directories, not to directories already existing!

2.2.8 *Error handling* view

This view gives you the opportunity to control what system-generated messages are displayed.

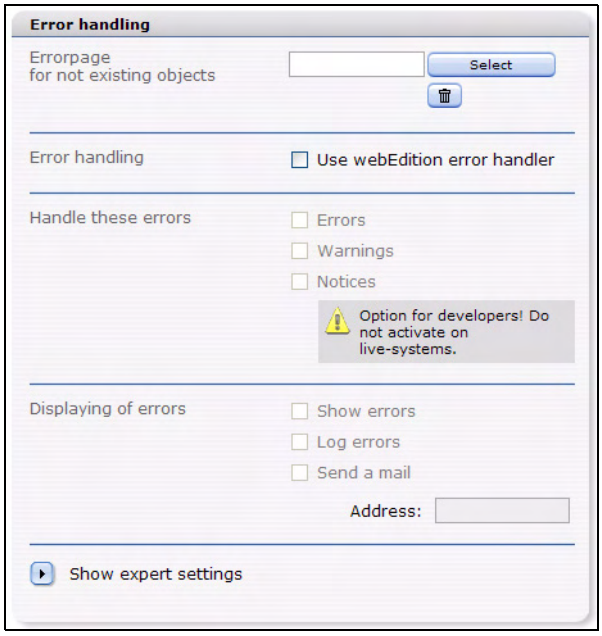


Figure 16 *Error handling* view

•*Errorpage for not existing objects*. Define a page to be displayed if a non-existant object is called.

•*Error handling*. If you want webEdition to display notices, warnings and/or errors, activate the error handling.

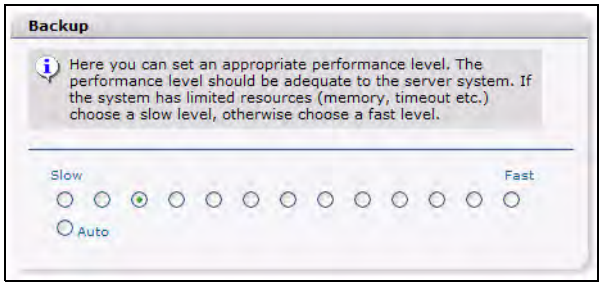
•*Handle these errors*.These options allow you to select what error messages the system is to treat.

- *Displaying of errors*. These options allow you to view error messages immediately on-screen, in an error log, and as an E-mail message.
- *Expert settings: Show debug frame*. You can turn the debug frame on or off in either normal or seeMode by selecting the designated checkbox. When activated, a frame where debugging information can be seen appears at the bottom of your screen. The debugging frame can be used to diagnose problems.

2.2.9 *Backup* view

Here, you can adjust the speed settings for your backup. Normally, this setting can be set to *Auto*. You should only change this setting, if you are experiencing problems with your backup.

Figure 17 *Backup* view

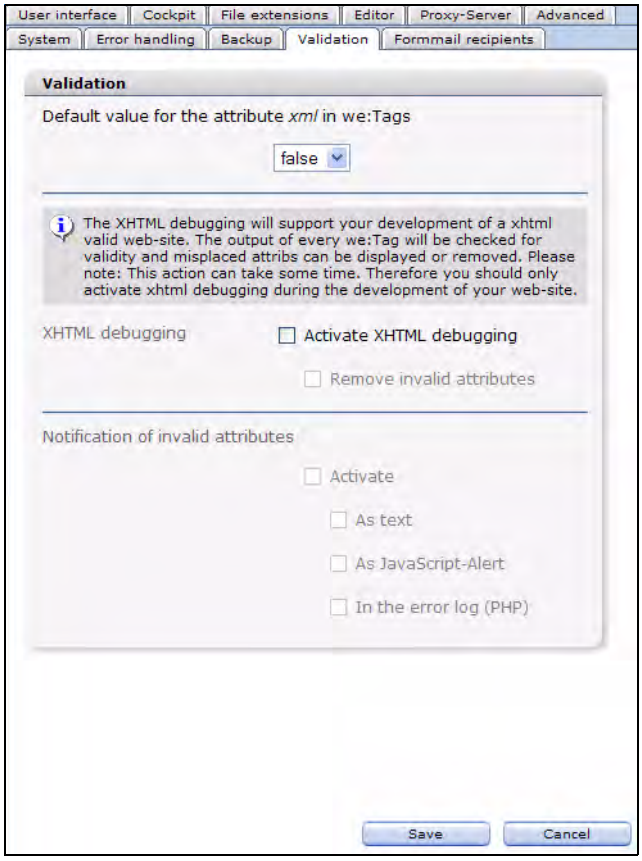


2.2.10 *Validation* view

The Preferences dialogue includes a *Validation* tab from which you can enter the *Validation* view.

The *Validation* view offers tools that are designed to assit you in the creation and maintenance of accessible or “barrier-free” Web sites.

Figure 18 *Validation* preferences



- *Default value for the attribute xml in we:Tags*: You can set the attribute xml=true globally for all tags (instead of having to do it for every tag). This command makes it possible to create valid XHTML-code.
- *XHTML debugging*: If you activate the checkbox, every we:tag will be checked for validity before output.

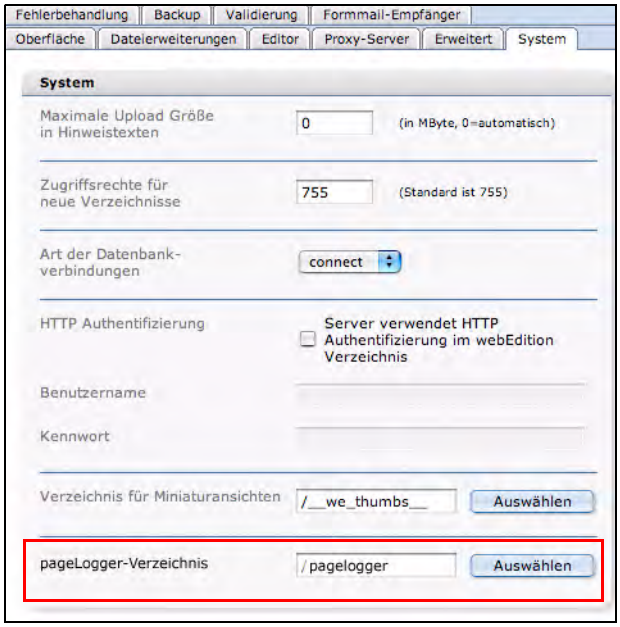
  Note: The debugging tool requires resources from your server and can affect perfomance of your site. Activate the debugging feature only during the testing of your site.

- *Notification of invalid attributes*: Activate this checkbox during testing to let the system display error messages either in an error log, a JavaScript alert box or in a text file.

2.2.11 System settings for pageLogger

The *pageLogger* area has been added to the *System* tab: *pageLogger-directory* (see Figure 19). Please select here the directory, where pageLogger is installed.

Figure 19 pageLogger-directory in system tab



As soon as the pageLogger directory is set here correctly, you are able to start pageLogger simply by clicking on *Modules* > *pageLogger*. Pleas note, that you have to restart webEdition for the changes to take effect.

> ***Note:*** For the *Single Login* feature to work correctly, a pageLogger user with the same name as the webEdition user has to be established. Furthermore, the user has to be marked as a *webEdition user* in the user preferences of *pageLogger*! For further information about webEdition users in pageLogger please consult the pageLogger documentation.

2.2.11.1 adding pageLogger to webEdition pages

To make the integration of pageLogger in webEdition-pages as easy as possibly, from webEdition Version 3.4.0.8 on, the new Tag `<we:pagelogger>` was created. This tag gives you the opportunity, besides using the code generated by the pageLogger Code Wizard, to easily integrate pageLogger into existing webEdition projects.

Further informationen about `<we:pagelogger>` can be found in the *webEdition Tag Reference* and the *pageLogger userguide*.

2.3 The webEdition menu structure

2.3.1 The *File* menu

The *File* drop-down menu has the following items (see Figure 20).

Figure 20 The *File* menu

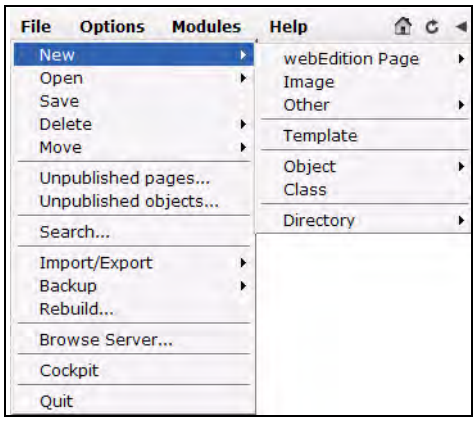


Figure 21 The *File* > *New* sub menu

•*New*. This item allows you to create a new webEdition page, to load an image, to create other documents (HTML page, Flash movie, Quicktime movie, JavaScript, CSS style sheet, and plain text documents, and others such as *.doc, *.pdf, *.zip, and *.txt files) or a new template (see Figure 21). Furthermore, you can create new directories in both explorer menu views—Documents and Templates.

•

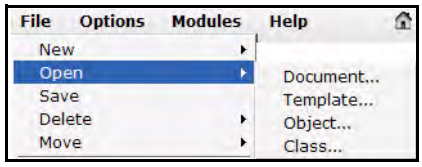
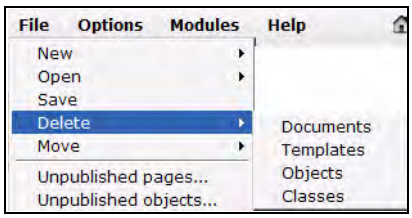


Figure 22 The *File* > *Open* sub menu

Open. This command allows you to open documents (webEdition pages, images, etc.) and templates (see Figure 22). If you have installed the Database/Object Module, this menu item also allows you to open objects and object classes.

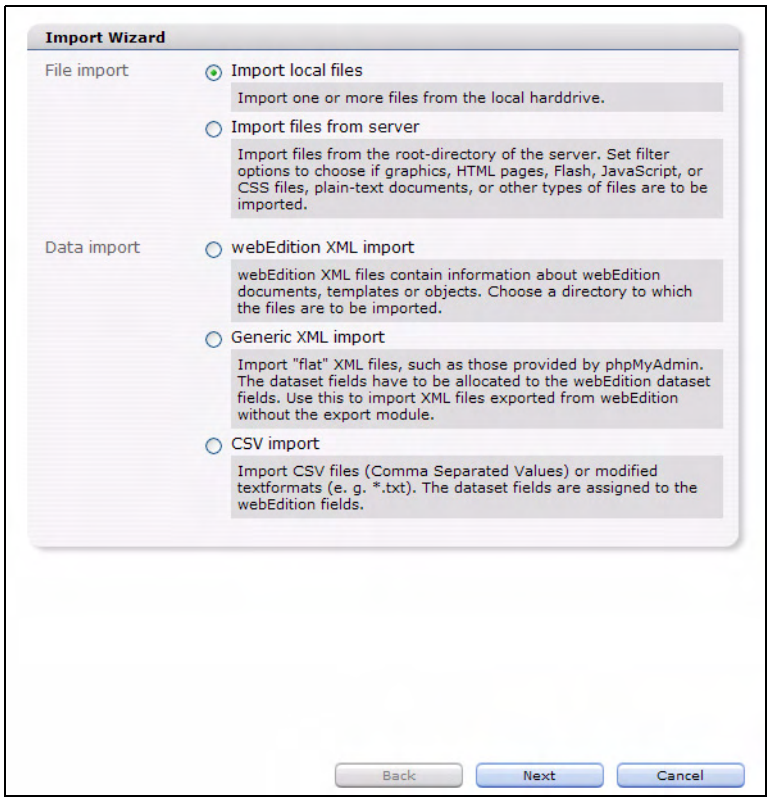
- *Save*. This item allows you to save the file that you currently have open in the edit view.
- *Delete*. This item allows you to delete a document or a template (as well as objects and classes if you have installed the Database/Object Module; see Figure 23).

Figure 23 The *File* > *Delete* sub menu



- *Unpublished pages*. This item opens a new window in which all unpublished pages are listed in a tabular format.
- *Unpublished objects*. If you have installed the Database/Object Module, you can use this command to display a tabular overview of all unpublished pages.
- *Search*. Click this item to open the webEdition internal search function.
- *Import/Export*. webEdition currently offers the following import functions (see Figure 24, "Import Wizard" on page 40):
  - *File import*. There are two file import options:
    - *Import local files*
    - *Import files from server*
  - Data import. There are two data import options:
    - *webEdition XML Import*. Imports webEdition specific XML files.
    - *Generic XML Import*. Use this generic extensible markup language (XML) import function to import flat XML files.
    - *CSV Import*. Use the comma separated values (CSV) import function to import csv files or modified text formats.

Figure 24 Import Wizard



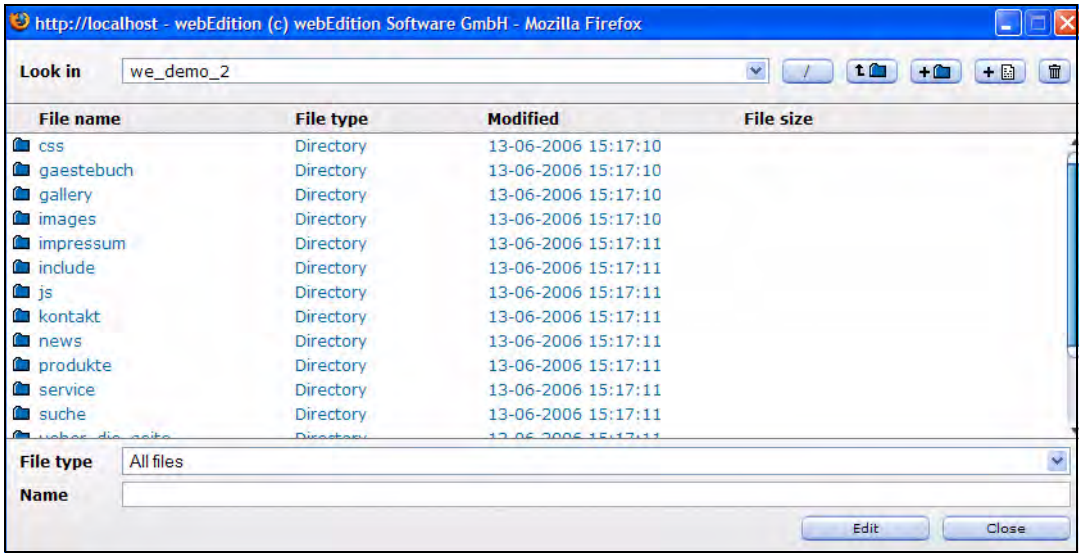
- *Rebuild*. This command allows you to re-write either all documents or those documents that meet specified criteria. The rebuild process is necessary to keep the links consistent whenever the linking structure changes, or if you have pages that are not generated dynamically and if you have edited a page which includes

elements and/or sections using the we:include tag. See Section 3.2.9, "Rebuilding static and dynamic Web pages" on page 65.

- *Backup*. The backup function has two parts:
  - The *Create Backup* command allows you to create a local backup of your host database. and/or to restore this image on your host.
  - The *Recover Backup* command allows you to restore your backup image on your host. For further details, see *The webEdition Installation, Update and Backup Procedures*.
- *Browse Server*. This command opens a file management window containing a complete overview of the root directory ("/") on your server. It displays all the directories and files created in webEdition in blue letters. Any other directories and files are displayed in black letters.

  The file manager helps you to work with and manage your files and directories. After selecting a file or directory, you can add, move or delete the item(s) using the file manager by clicking one of the icons at the top-right of the screen. (see Figure 25).

Figure 25 The file manager



- *Cockpit*. Returns you to the *Cockpit* menu (see Section 2.1.2.1, "The Cockpit" on page 27).
- *Quit*. Select this command to log out of webEdition.

2.3.2 The *Options* menu

The *Options* menu has the following items (see Figure 26, "The Options menu" on page 41):

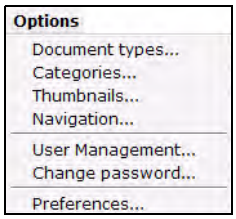


Figure 26 The *Options* menu

•*Document types*. This command allows you to create a new document type.The *Document types* feature allows you to group documents with similar characteristics, or to assign characteristics to a new document that are similar to existing documents. You can also

select an existing document type and delete it or modify its properties, which might entail assigning a template to it or specifying default values. See Section 3.6.1, "Document types" on page 89.

- *Categories*. This command allows you to create a new category or to delete categories you no longer need. The *Categories* feature offers an alternate method of generating page summaries, and of organizing your documents. See Section 3.6.2, "Categories" on page 92.
- *Thumbnails*. Here you can apply the settings for thumbnails.
- *Navigation*. Open the navigation tool.
- *Change user name/User management.* This command allows you to change your user name. If the User Management Module is installed, *User Management* appears in the menu. See Section 3.6.3, "Changing user name and/or password" on page 93.
- *Change password*. This item allows you to change your password. See Section 3.6.3, "Changing user name and/or password" on page 93.
- *Preferences*. Use this item to modify the default webEdition preferences. See Section 2.2, "Preferences" on page 29.

2.3.3 The *Modules* menu

The *Modules* menu lists all webEdition modules installed in your version of webEdition (see Figure 27, "The Modules menu" on page 42). You can open a module by clicking on the corresponding menu item.

Note: The *Modules* menu will vary depending on the modules you have installed.

Figure 27 The *Modules* menu

For further details about a specific webEdition module, see the supporting documentation at http://www.webedition.de/english/downloads/Documentation.html

2.3.4 The *Help* menu

The *Help* menu contains the following items (see Figure 28, "The Help menu" on page 42):

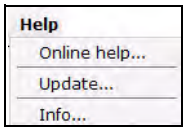
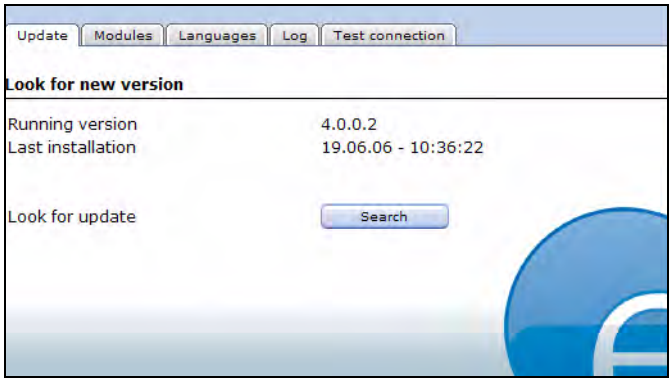


Figure 28 The *Help* menu

•The *Online help...* menu contains an extensive help program. You must be connected to the internet. The help program is on our server.

- *Update*. Opens the Update window, which contains the following tabs

Figure 29 Update window



— *Update*. Here you can check which version of webEdition you have installed. The version number appears at the top of the view. You can use the Search button to check if there is a newer available version. If so, you can update to a newer version using the online installer. In addition, all new modules are installed using this menu item.

— *Modules*. Open this tab and follow the instructions to install purchased webEdition modules.

— *Languages*. Open this tab to search for and install additional languages

— *Log*. Shows you logged informations.

— *Test connection*. In this view you can test your connection to our update/module installtion servers.

• *Info*. This item contains information on webEdition, and the version you are using.

2.3.5 The explorer menu

The explorer menu appears on the left side of the webEdition main screen, and it displays the contents of your Web site as a series of folders and files. Adjacent to the explorer menu are either two or four vertical tabs, depending on whether or not you have installed the Database/Object Module (see Figure 30, "The explorer menu" on page 43). When you click on a vertical tab, you can view the corresponding editing view in the explorer menu:

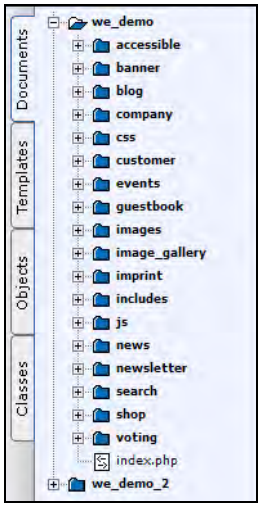


Figure 30 The *explorer* menu

•*Documents* tab. This editing view shows a list of the available documents (Web pages, style sheets, images, etc.) and the document directories

•*Templates* tab. This editing view shows all the templates

•*Objects* tab. This editing view (visible only if you have installed the Database/Object Module) shows a list of available objects

•*Classes* tab. This editing view (visible only if you have installed the Database/Object Module) shows a list of available object classes

Documents, templates or objects displayed in red in the explorer menu have been saved, but have not yet been published, while

those displayed in black have been both saved and published. See also Section 3.2.2, "Saving, “unpublishing”, and deleting documents" on page 60.

By clicking on one of the view icons, you can open, change or publish a desired file in edit mode. By double-clicking one of the entries in the *Documents* view, you will launch a new browser window showing you the document as it will be displayed online.

2.3.5.1 Identification numbers (IDs)

If you pass the mouse pointer over the icon of an individual document or template, a number appears below the explorer menu. The same number appears on the screen below your mouse pointer (see Figure 31).

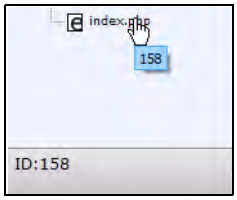


Figure 31 webEdition ID numbers

IDs are unique numbers allocated to a document or a template. webEdition uses the ID to access the document internally. The ID allocated to a document is needed for some we:tags (such as we:a or we:url). The document ID is designed to make it easier for you to link documents.

2.3.5.2 Explorer menu icons

The various file types used in and with webEdition are identified in the explorer menu by the distinctive icons:

Directories and sub directories are identified by folder icons:

Figure 32 directory icon

webEdition pages (.php and .html files) are identified by “e” icons:

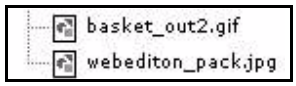
Figure 33 webEdition page icon

Templates, JavaScript, CSS (.tmpl, .js, .css files) appear thus:

Figure 34 Template icon

Images (.gif, .jpg, jpeg, .png files) appear thus:

Figure 35 Image icon



Movie files (.mov, .swf) appear thus:

Figure 36 movie icon

Plain text (.txt) files appear thus:

Figure 37 Plain text icon

If you have installed the Database/Object Module, the following icons appear in the *Objects* and *Classes* views:

- objects

Figure 38 Objects icon

- object class icon

Figure 39 Object class icon

2.4 Search function

2.4.1 Simple search

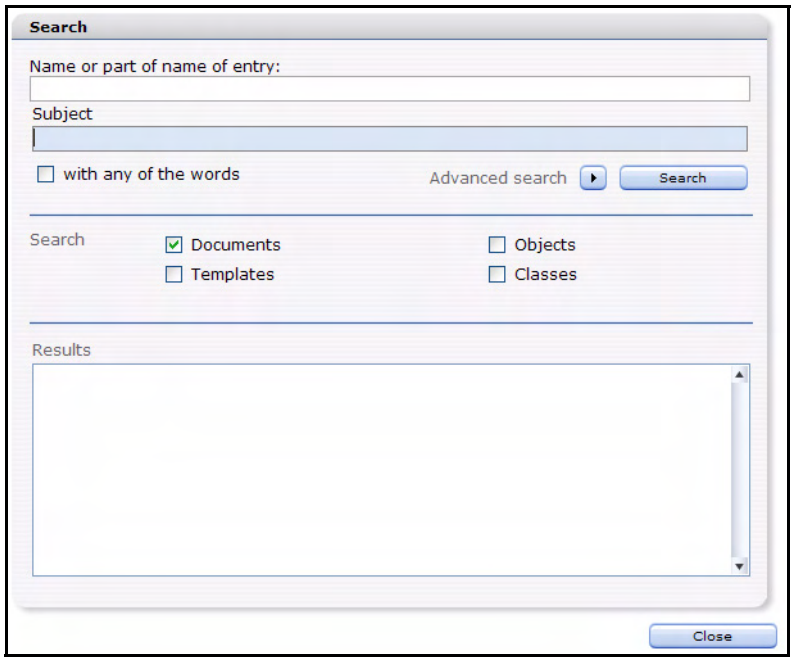
You can reach the webEdition internal search function at any time the input field on the bottom left, below the file tree.Simply enter the search string in the field and press the Return button or click on the magnifying glass icon.

Figure 40 Input field for search

Note: In this case the search refers to the type selected in the main view: Document, template, object or class! If you are in the documents view, only documents are browsed.

The result of your search is indicated immediately. Another possibility is the use of the menu item *File > Search...* . The search window is opened.

Figure 41 The *Search* window - simple search



Note: The results from the input field are displayed in this window

In the field Subject you can enter the search string to be searched. According to your choice of the check boxes, all documents, objects, templates or classes are searched. In the results area, the hits are displayed. The suitable document,object, class or template is opened in the webEdition main window by double click on a result. The search window remains opened, on this occasion, so that you can continue to search comfortably.

Figure 42 List of results

In the field *Name or part of name of entry* you can enter a search string to search for documents, templates, objects or classes according to their naming within the webEdition fille tree. The operation occurs analogously to the search for a *Subject*.

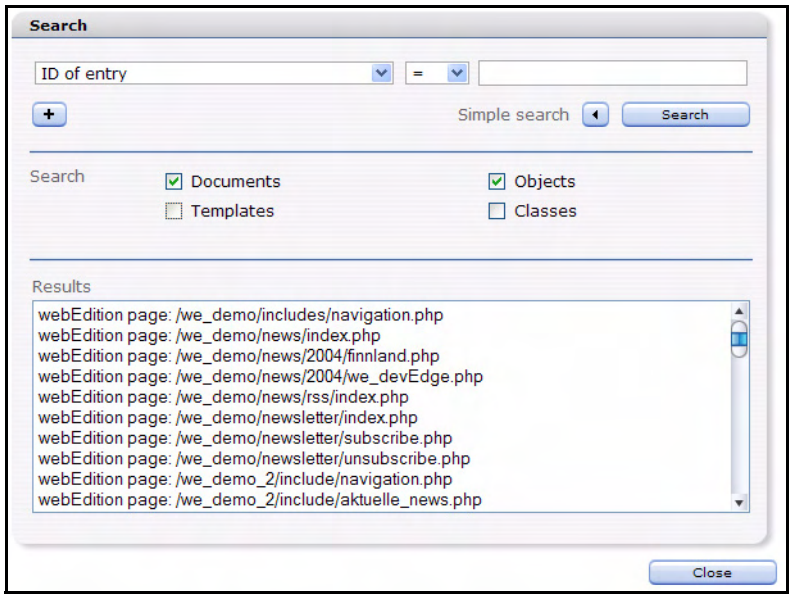
2.4.2 Advanced search

The requirements for the filtering of the search results may rise. Hence, there is the advanced search. You access the advanced search with a click on the triangular button beside *Advanced Search* in the search window

Figure 43 Activating advanced search

Now the advanced search possibilities can be controled very specifically via dropdown menus, logical operators and input fields and combinations thereof.

Figure 44 The *Search* window- advanced search



The search can be controled by logical operators:

Operator:	**bedeutet:**
=	equal
<>	not equal
<	lesser than
<=	lesser or equal
>	greater than
>=	greater or equal

The items in the dropdown menu:

- *ID of entry*: Searches for document, template, object or class (depends on selected checkbox) with given ID
- *Path of entry*: Searches for document, template, object or class (depends on selected checkbox) in specified path
- *ID of parent entry*: Searches below the “directory“ with the given ID
- *Text of entry*: Search for the name of the entry
- *Template of entry is*: Search for **documents** based on the template with the given ID!
- *Type of content*: Searches within the selected content type (img/*; /html)

- *Document-type*: Search documents based on the given document type
- *ID of object class*: Search objects based on the given class.

Note: You can also search from the search field for e.g. a document with certain id: enter for this in the search field exp:id = ” id of the document ”. It is searched for a document with id if you are in the documents view! (Document tab)

2.5 webEdition control elements

To the right of the drop-down menus there are the following navigation control elements:

Figure 45 webEdition navigation control elements

- *Home*. If you click on the symbol which looks like a house, the *seeMode start document* opens. You can define the *seeMode start document* in *Options > Preferences*.
- *Reload*. Click the icon adjacent to the Home icon to reload the Web site.
- *Back.* Click on the left-pointing arrow icon to navigate to the page opened last.
- *Forward*. Click on the right-pointing arrow icon to navigate to the next open page.

2.6 Viewing modes: standard and Super-Easy-Edit Mode

There are two viewing modes available in webEdition: normal mode and Super-Easy-Edit Mode (seeMode)

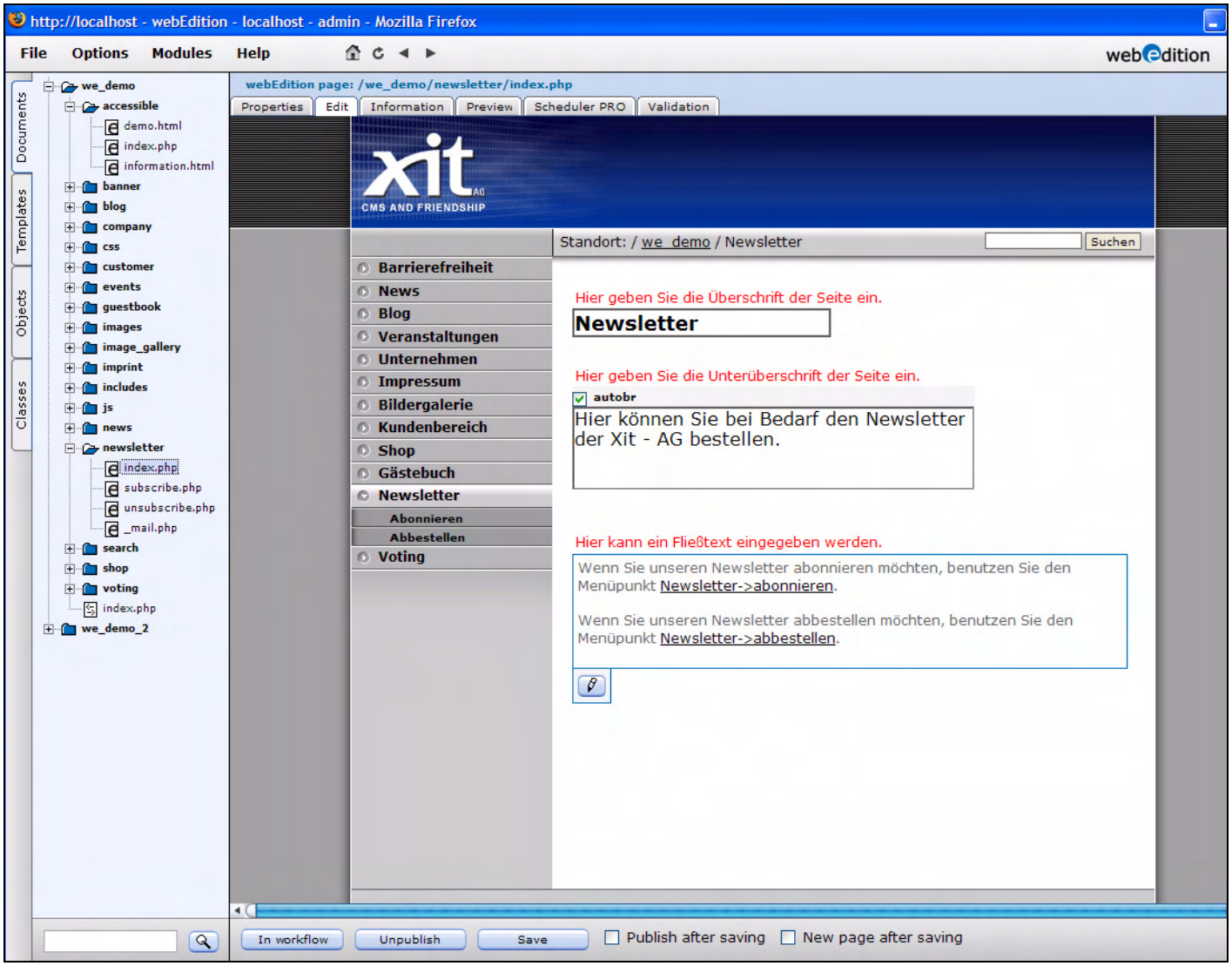
2.6.1 The editing screen in standard mode

The editing screen is where you can create, edit, preview your individual webEdition documents and templates. If you are an editor and you click *File > New > webEdition Page > Empty Page*, the *Documents* editing screen opens. Four tabs are visible, a fifth is visible in the document editing view if you have installed the Scheduler Pro Module (see Figure 46, "The webEdition Edit view (normal mode)" on page 49). Likewise, if you are an administrator and you click *File > New > Template*, the *Templates Edit* view opens. Each tab, when clicked, opens a specified view.

- *Properties*. This view allows you to assign all the necessary properties to the file such as name, directory, document type, etc.
- *Edit*. This view displays the *Edit* view, document you wish to edit in edit mode. You make all your entries to the page in this mode.
- *Information*. This view displays all relevant information on the current document.

- *Preview*. This view allows you to check how your page will actually appear in the browser.
- *Scheduler PRO*. If you have installed the Scheduler Pro Module, when you are working with a document, you can use this view to access the settings for the module.

Figure 46 The webEdition Edit view (normal mode)



The editing screens and the available views are different for documents and templates. Each will be discussed, in turn, in subsequent chapters.

2.6.2 The editing screen in Super-Easy-Edit Mode

In contrast to the normal editing mode, which serves the administration of the entire Web site, the seeMode is ideal for editing content.

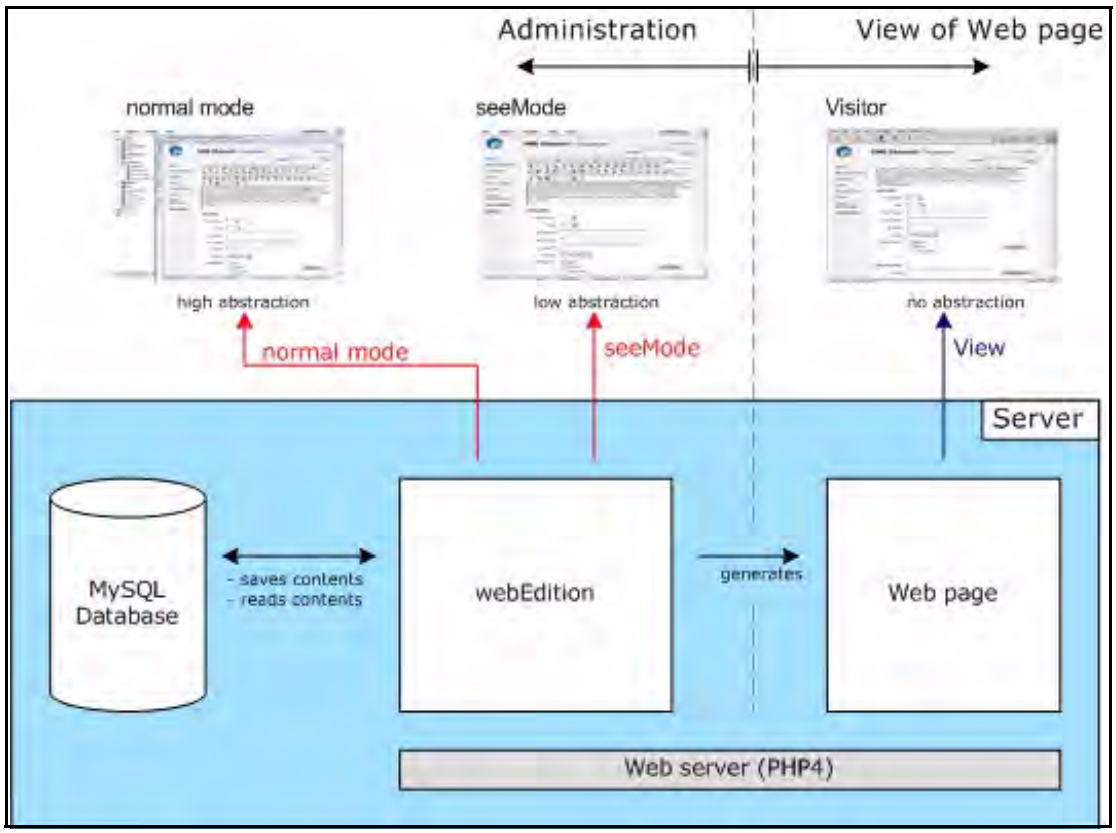
The seeMode is characterized by simplified navigation and a removal of the more complex functions that are better suited to the skills of an administrator. The seeMode is designed for the simple maintenance of existing Web site content.

> ***Note:*** The seeMode feature in webEdition 3.0 cannot be used with Web sites that use HTML frames.

When in Normal mode, an editor must select each of his/her documents from a file tree; in seeMode an editor can navigate to all the documents for a Web site using the navigation of the web page that they must edit. In contrast to working with a file tree, which presupposes a certain understanding of where and how documents are stored on a Web server, the desired document can be accessed easily and directly. An editor using seeMode does not need to know how the documents for a Web site are distributed on the server; when an editor logs in, webEdition can be configured so that the Web page that they have to maintain automatically appears. The editor can navigate through the Web site and modify the contents of the associated pages using a WYSIWYG editor and then published the revised pages. An editor can perform all of the same tasks in seeMode as in normal mode, except template and class editing.

Figure 47 shows how the view of an editor, administrator and end visitor will perceive a Web page.

Figure 47 What editors, administrators and visitors see



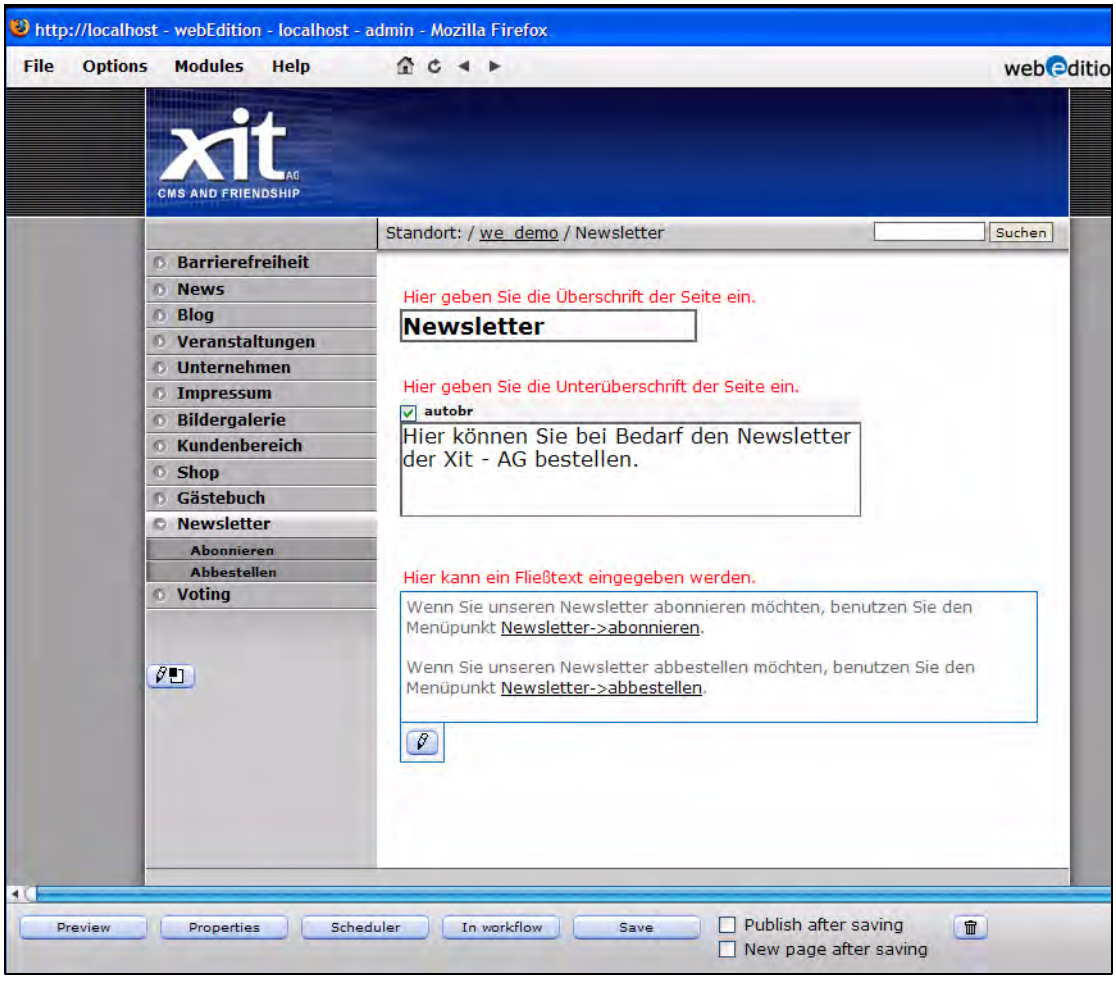
The first time you log into webEdition in seeMode, you will see the following message (see Figure 48), asking whether you wish to choose a default web page which will open each time you log in to seeMode:

Figure 48 The *No start document* warning

If you click *OK*, the preferences screen opens in the *User interface* view. In the *seeMode start document* area you can choose a start document by clicking the *Select* button and browsing the server for a document.

Thereafter, every time you log into webEdition in seeMode, the page you have selected will automatically appear on screen. This page is your seeMode home page. See, for example, Figure 49.

Figure 49 The webEdition editing screen (seeMode)



2.6.2.1 seeMode user interface

The menu structure and functions for the seeMode are the same as what is described in Section 2.3, "The webEdition menu structure" on page 38, with the following exceptions pertaining to navigation and file creation:

2.6.2.1.1 explorer menu The file tree that one sees in Normal mode does not appear in seeMode. All navigation can be carried out using the links inherent in your seeMode start page. You do, nevertheless, have the option of opening files from the server by using the *File* > *Open* menu.

2.6.2.1.2 *File* menu All functions are available to the user. If you have installed the User Management Module, user activities can be restricted (see Section 2.6.2.1.4, "Interactions with the User Management Module" on page 52).

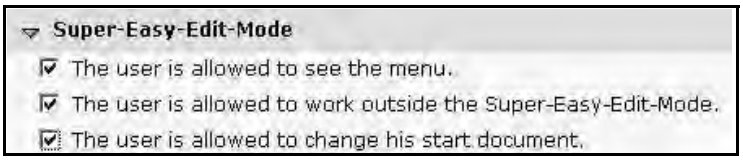
2.6.2.1.3 *Home* menu When you are editing your web pages in seeMode, you can click the *Home* menu to return to your designated seeMode start page.

For details on how to edit documents while using the seeMode, see Section 3.8, "Editing documents in seeMode" on page 106.

2.6.2.1.4 Interactions with the User Management Module The User Management Module gives administrators additional flexibility in designating what a user will see when he/she logs into webEdition using seeMode. There are three control features for seeMode, which are accessed from the User Management *Permissions* view (see Figure 50):

- *The user is allowed to see the menu*. If this checkbox is not activated, for a given user, that person will see only the *Quit* and *Home* menus. Hence the person will be limited only to editorial tasks associated with an existing document.
- *The user is allowed to work outside the seeMode*. Here an administrator can restrict access to files that were not created in webEdition.
- *The user is allowed to change his/her start document.* If this checkbox is not activated, the user will only be able to modify the seeMode start page (and its internally linked pages).

Figure 50 Permissions for the seeMode in the User Management Module



3 Working with documents – the role of the editor

The webEdition content management system differentiates between the role of the editor and the role of the administrator. Each role requires a different skill-set. Moreover, the tasks assigned to each role are governed by different webEdition user permissions.

This chapter describes the activities that are performed when working with webEdition documents. This role is normally carried out by a Web page editor. These tasks include:

- creating new documents
- saving/parking/deleting documents
- modifying/updating documents
- importing files
- working with other documents: graphical and non graphical files
- creating new document directories
- defining categories
- rebuilding static and dynamic Web pages
- changing user names and/or passwords
- editing preferences
- the navigation tool

By "documents", we mean all types of files that are used in the presentation of a Web page. Such files can include webEdition pages, normal HTML pages, cascading style sheets, JavaScript pages, images and Flash movies.

> ***Note:*** This manual always refers to the sample pages supplied with the software. Sample pages can be accessed from the we_demo directory, found in the webEdition explorer menu.

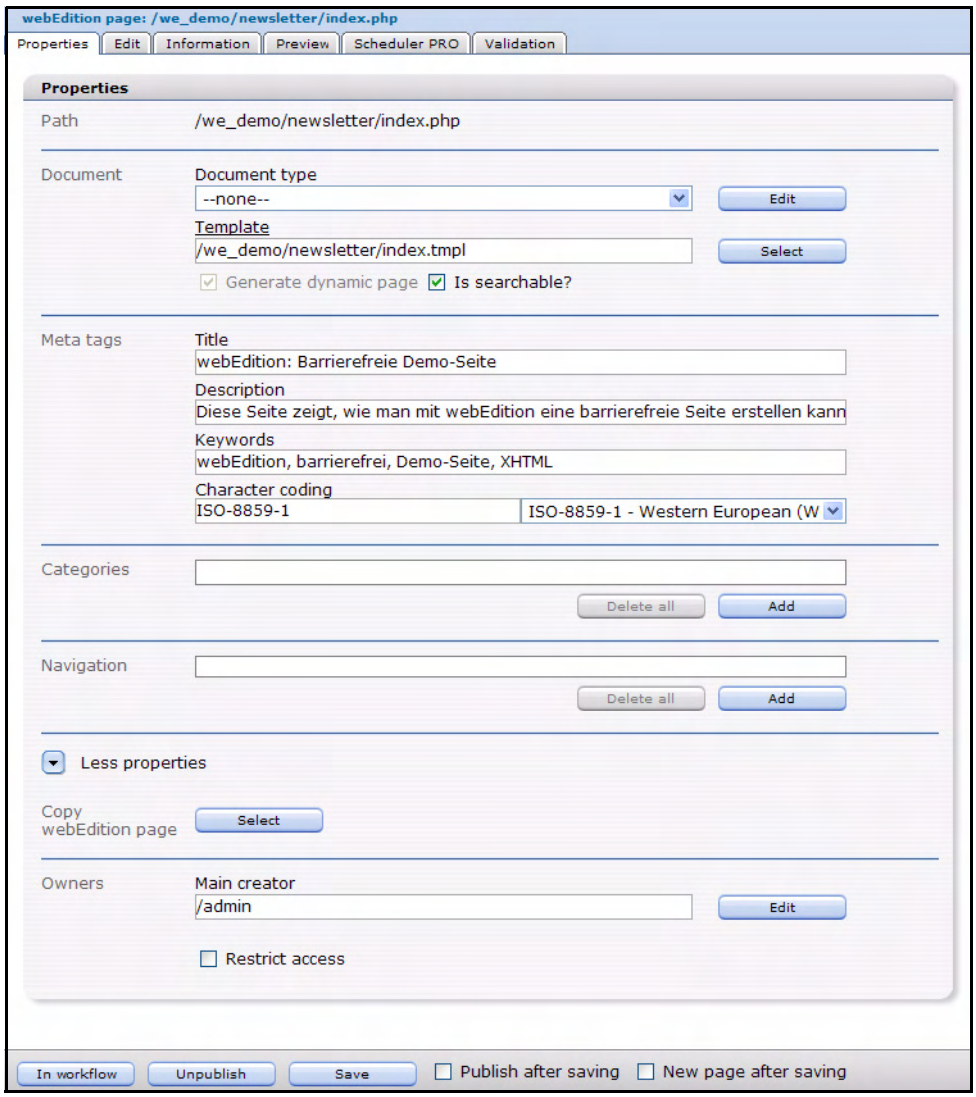
3.1 The four views when working with documents in normal mode

In Chapter 2, we discussed how the editing screens and the available views are different depending on whether you are working with a document or a template. The following sub-sections describes each of these views as they pertain to documents.

3.1.1 The *Documents Properties* view

You use the *Properties* view to define the name, type, location, and ownership of your webEdition documents. The *Properties* view has the following areas: *Path*, *Document*, *Meta Tags* and (under the *More properties* button) *Categories*, *Owners* (see Figure 51).

Figure 51 The *Documents Properties* view



- *Path*. This area allows you to enter the name of the document, its extension and the directory in which it is to be saved. Select the directory by clicking the *Select* button and using the file manager. To avoid input errors, the directory cannot be entered by hand.

  The extension determines how the server will react to the document. The extension for static pages is either .html or .htm, whereas the extension for dynamic pages is .php.

  Note: If you allocate an .html extension to a dynamic document, it will not execute and the viewer of the document will be able to see the source code, unless your administrator has configured your server to have the PHP parser execute documents with an .html or .htm extension. If you are using other script

languages such as ASP or JSP in your templates, always save pages as "static" pages and select the appropriate extension (for example .asp for ASP pages)

- *Document*. In the *Document* area you can select the document type and the template on which you wish to base the new page.

  If you choose to select a document type, your selection will be limited to the template allocated to this document type. For additional information on document types, consult Section 3.6.1, "Document types" on page 89. If you do not select a document type, you will be able to use any of the available templates. The checkbox located at the bottom of the *Document* area allows you to determine if the page should be created dynamically. If you do not activate the checkbox, the page will be created statically. You can find more information on the difference between static and dynamic pages in Section 3.2.9, "Rebuilding static and dynamic Web pages" on page 65.

  To make a document searchable, activate the *Is document searchable?* checkbox.

- *Meta Tags*. In the *Meta Tags* area, enter in the *Title* field the title you want to appear in the browser bar when the page is viewed by an end user. This is equivalent to an entry in the <title> HTML tag as long as the we:tag is in the template. In the *Description* field, enter the description of the Web page that you wish to be displayed by, for example, a search engine. This is equivalent to an entry in the <description> meta tag, as long as the we:tag is in the template. In the *Keywords* field, enter the keywords that characterize the content of the document. These words will be found by search engines and other such programs. This is equivalent to an entry in the meta tag <keywords> as long as the we:tag is in the template. You may also define your character coding for your document here.

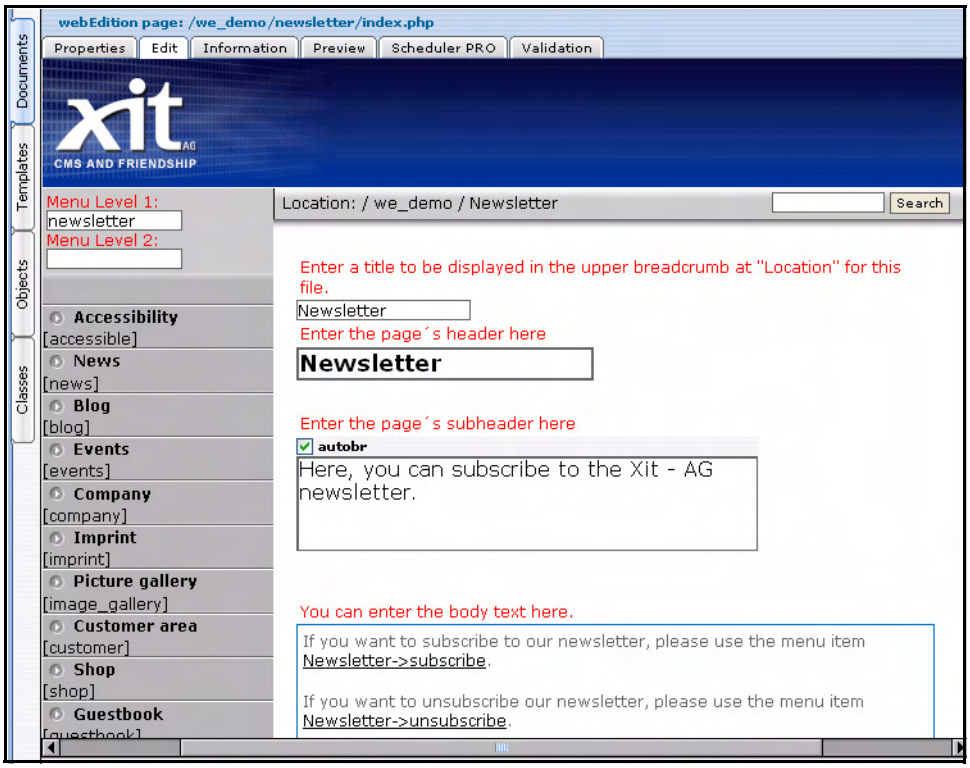
- *Categories*. In the *Categories* area, you can select the category to which you want your document to belong. To do so, click on the *Add* button to make a selection from the available categories. For additional information on categories, please consult Section 3.6.2, "Categories" on page 92.

- *Navigation* In this area you can add this document to the navigation (if the naviagtion of your site is created and administered by the navigation tool).

3.1.2 The *Documents Edit* view

You use the *Edit* view to create or modify the content of your document by entering text, creating link lists, adding images, etc. (see Figure 46, "The webEdition Edit view (normal mode)" on page 49). With the exception of the areas that still have to be edited, the layout you see in the *Edit* view shows how this page will look on your Web site. The areas in which you can work appear as entry fields and are pre-defined by the template(s) created by your webEdition administrator. (More information on how this is done can be found in Section 6, "The role of the administrator – creating templates".)

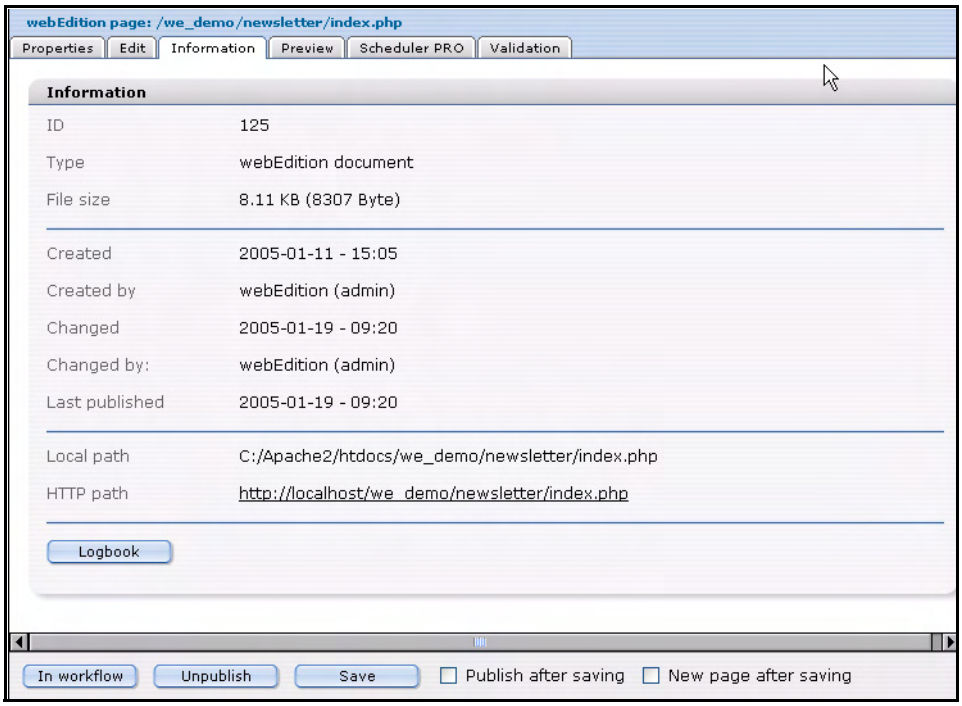
Figure 52 The *Documents Edit* view



3.1.3 The *Documents Information* view

The *Information* view provides you with read-only information about your document, such as the file type, the size of the file, the date created, the date when it was last modified and when it was last published. This view also provides information on the local host path where the file has been saved and the HTTP path (the URL) where the page can be accessed online (see Figure 53).

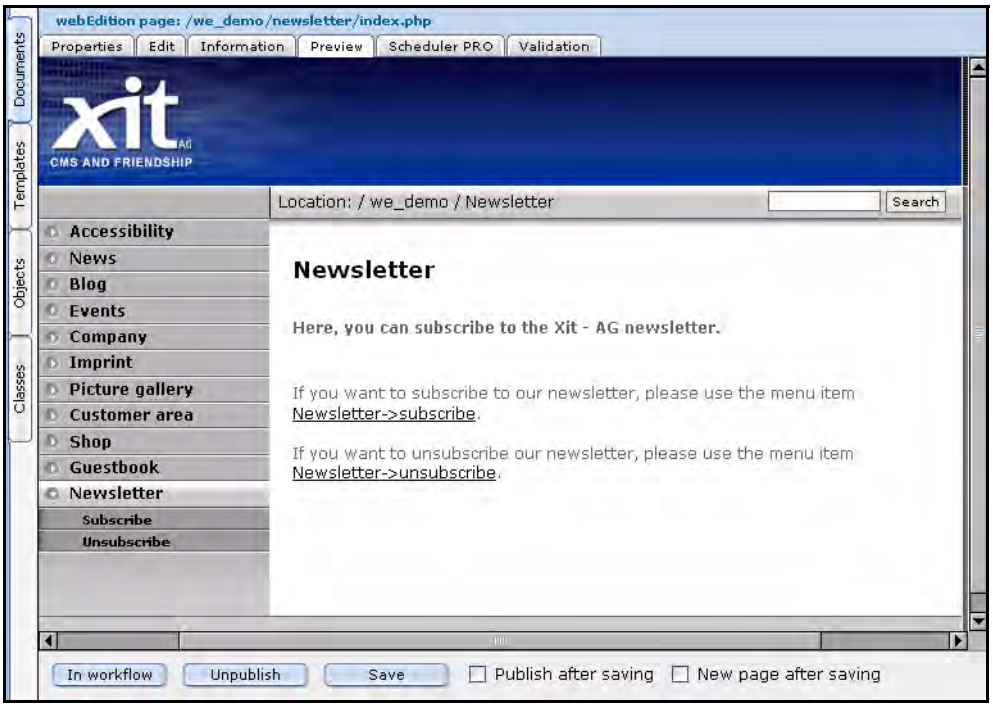
Figure 53 The *Documents Information* view



3.1.4 The *Documents Preview* view

You can use the *Preview* view to display how your finished page will look online when viewed in a browser. Use this feature to check the contents of the page before saving or publishing it (see Figure 54).

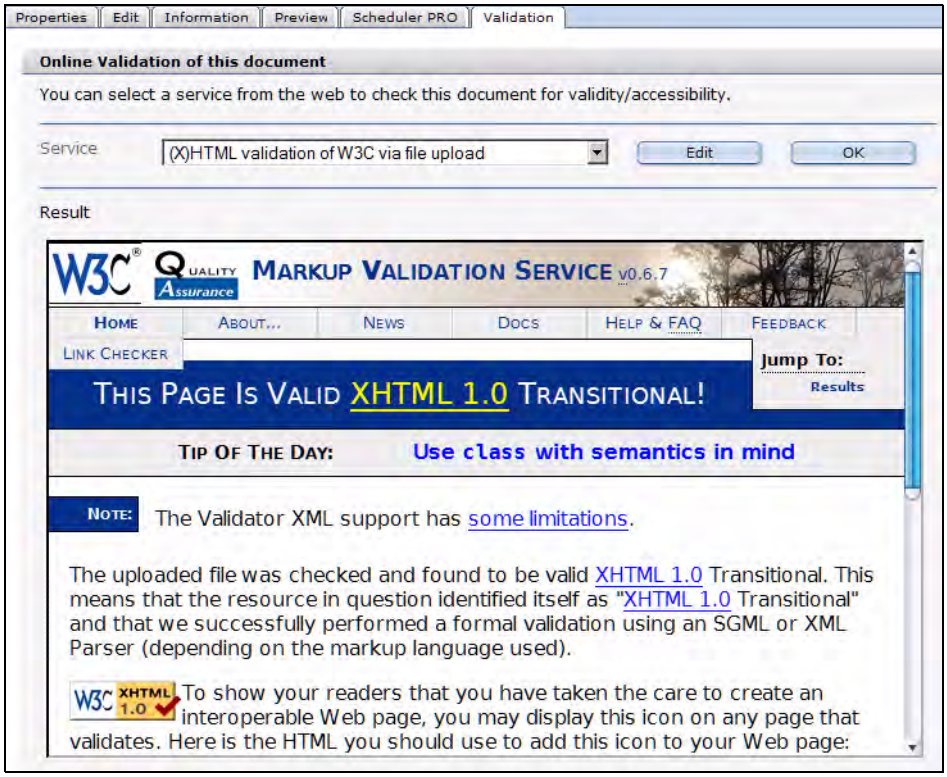
Figure 54 The *Documents Preview* view



3.1.5 The *Documents Validation* view

Click on the *Validation* Tab to access the validation view.

Figure 55 The *Documents Validation* view

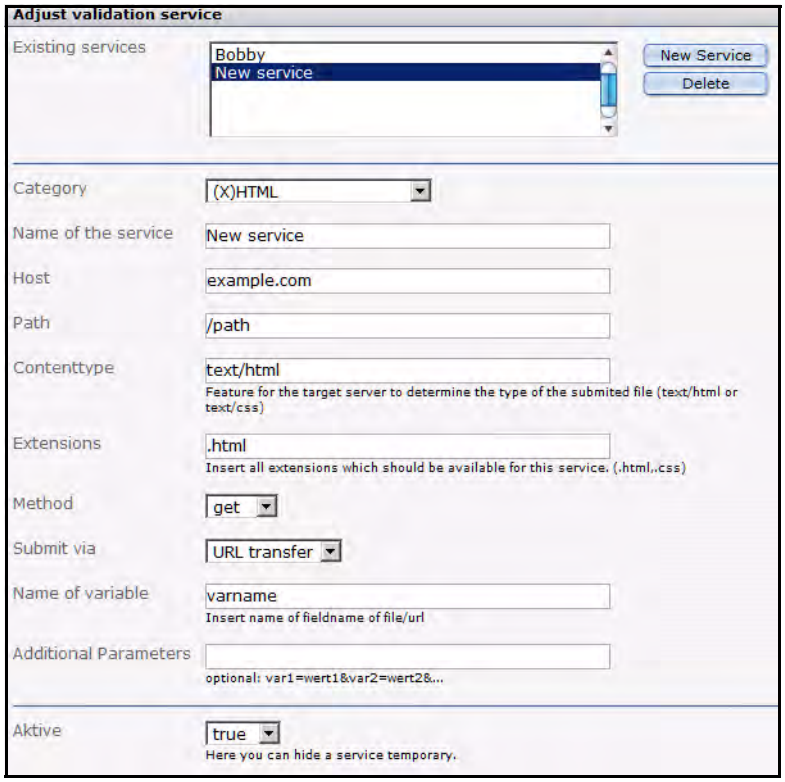


On the Internet there are public services intended to help you to verify the validity of your code and the accessibility of your web page. webEdition includes the *Validation* view so you can validate your webEdition documents for barrier-free compliancy from within the CMS system either via URL transfer or file upload. The response of the service is displayed unmodified in webEdition (see Figure 55 on page 58).

To have your page checked by one of the preset services, select the service from the *Service* dropdown menu and click *OK*. In Figure 55, you can see the response for a successful validation according to the rules of the World Wide Web Consortium (W3C).

You can add other services by clicking on the *Edit* button and then adding a *new service* (see Figure 56).

Figure 56 Adding a new service



You can subscribe to additional services such as:

- Validome (http://www.validome.org)
- Accessibility Bobby (http://bobby.watchfire.com)
- Cynthia Says (http://www.contentquality.com/)
- Checky (http://checky.sourceforge.net/) offers a list of validation services.

3.2 Working with webEdition pages

3.2.1 Creating a new webEdition document

Use the following procedure to create a new document.

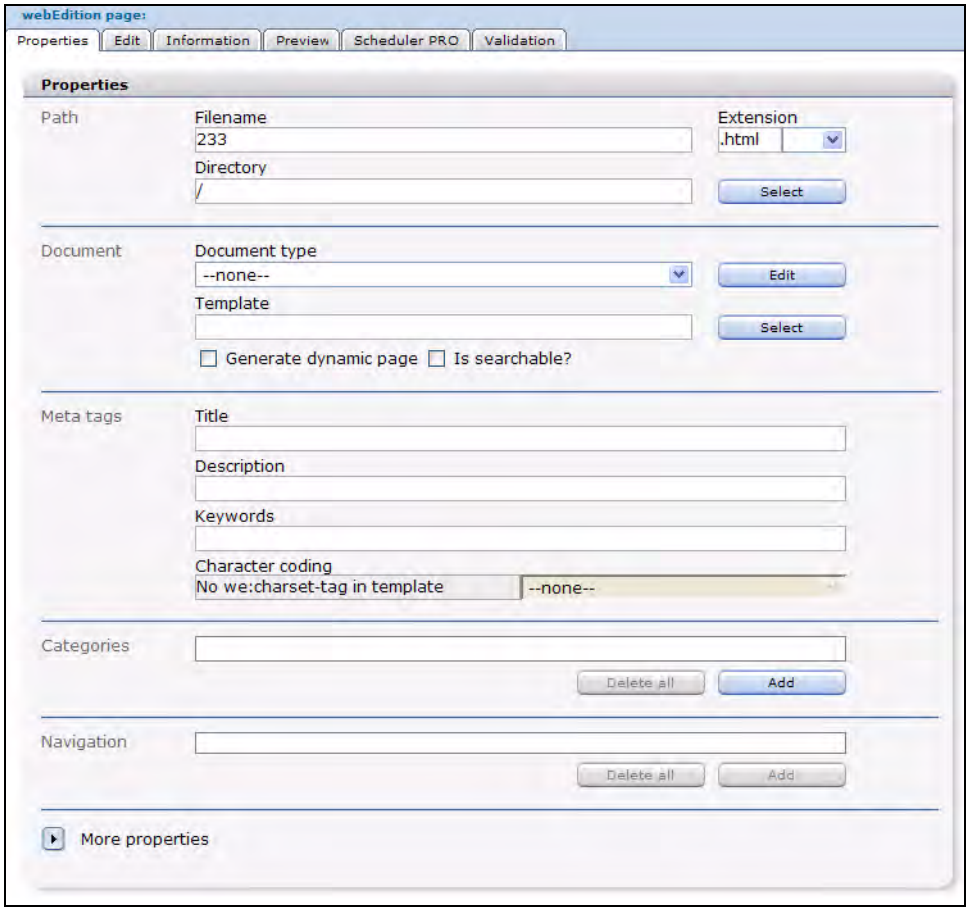
Procedure 1 Creating a new document

At the webEdition main screen

1 Select *File* > *New* > *webEdition Page* > *Empty Page*.

The webEdition Page Properties view appears (see Figure 57).

Figure 57 Creating a new document: *Properties* view



2 In the *Path* area of the *Properties* view, enter a name for the new document in the *Filename* field.

3 In the *Path* area of the *Properties* view, choose a directory for the new document by clicking the *Select* button adjacent to the *Directory* field.

   a Navigate through the directory menus until you find the location for the new file.

   b Click on the name of the target directory, then click *OK*.

4 Click the *Save* button located at the bottom of the screen.

   webEdition opens a read-only information box informing you that the new file has been successfully saved.

5 Click the *OK* button.

   The newly created file is now listed in red letters in the explorer menu and you can now start editing your new document. When a page is displayed in red, this indicates that it has been saved but not yet published.

6 You have completed this procedure.

Note: This procedure creates an “empty“ page. The design is applied by the use of templates. To create a new document similat to an existing one, just use the *Copy webEdition page* option to create a document based on an existing one.

3.2.2 Saving, “unpublishing”, and deleting documents

After creating or modifying a document, you must save your files.

3.2.2.1 Saving and publishing a page

When working with a document, you can save a file while in any of the four views (*Property*, *Edit*, *Information*, *Preview*). webEdition differentiates between two situations: saving a new page and saving an existing page that has been changed.

- Saving a new page. You will be given the following options when saving a new page:
  - — *Without publishing it*. If you click the *Save* button without activating any of the check boxes, the page will be saved in the system, but will not yet be published. It will appear in the explorer menu in red.
  - — *Publishing it*. Before saving, activate *Publish after saving* and then click the *Save* button. The page will be saved and published at the same time so that it can be accessed from outside webEdition. Published pages appear in the explorer menu in black font.
  - — If you wish to have a new, empty document of the same type available for editing immediately after saving a document, activate the *New page after saving* checkbox.
- Saving an existing page.
  - — *Without publishing it*. If you change an existing page and want to save it without changing the published page, make sure that the *Publish after saving* checkbox is not activated, then click *Save*. The document will appear in the explorer menu in blue, indicating its new status in "temporary storage" without being published—the old page will remain published.
  - — *Publishing it*. If you wish to publish an existing page immediately after changing it, or after having saved it temporarily, activate the *Publish after saving* checkbox and click on the *Save* button. If the page has been in temporary storage, the blue letters denoting its status will then become black in the explorer menu. You can check whether the page has been published by double-clicking on the document in the explorer menu. If it has been published, the new browser window will open displaying the page. If you try to double-click on a page that has not been published (red), a new browser window will appear informing you that the document has not been published.

    If you wish to have a new, empty document of the same type available for editing immediately after saving a document, activate the *New page after saving* checkbox.

3.2.3 "Unpublishing" a file

If you wish to “unpublish” (or “park”) a published page without delete it, you can use the *Unpublish* function to place the file in temporary storage. Click on the page in the explorer menu to bring up the page in *Edit* mode. Instead of clicking *Save*, click *Unpublish*. The page will no longer be live and will appear in red letters in the explorer menu. If you wish to re-publish this page, just activate the *Publish after saving* checkbox and click the *Save* button.

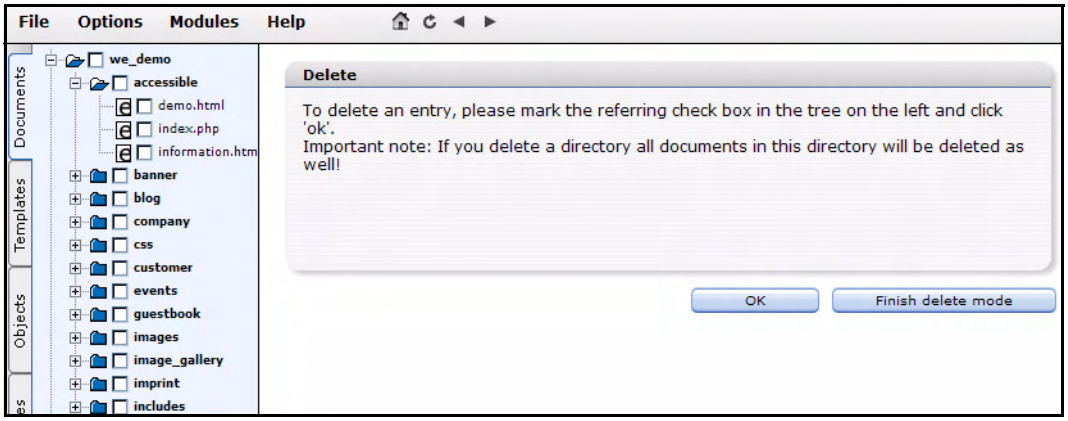
Note: If a document has already been published, you can only change whether it is generated statically or dynamically by unpublishing it, making the change and re-publishing it.

3.2.4 Deleting documents, templates, objects, or classes

In this section, by way of example, we describe the deletion process for a document. To delete a template, object, or class, you proceed in a manner similar to the steps described here.

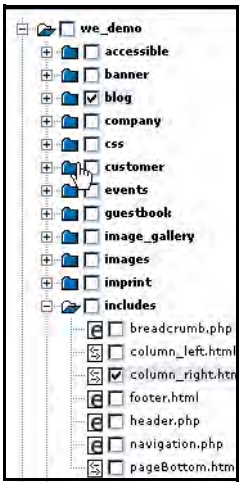
You can delete a document by clicking *File* > *Delete* > *Documents* (see Figure 58).

Figure 58 Deleting a document



A checkbox appears beside each document and directory in the explorer menu (see Figure 59).

Figure 59 Marking files for deletion



Activate all of the checkboxes for the entries that you would like to delete.

CAUTION

When you delete a directory, all documents and subdirectories contained within it will be lost. In order to exit from the delete mode, click the *Cancel* button. Click on the *OK* button to delete the file.

.

3.2.5 Overview of all unpublished pages

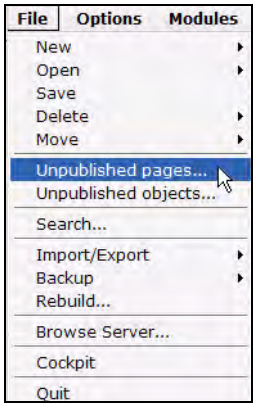
You can access an overview of all unpublished pages or objects, in two ways:

- After logging in to webEdition, you can go directly there by clicking on the *Unpublished pages* (or *Unpublished objects*) button on the *Quickstart* screen (see Figure 60).

Figure 60 Quickstart-unpublished pages

- You can also select *File* > *Unpublished pages* (or *Unpublished objects*) from the *File* drop-down menu on the webEdition main screen (see Figure 61).

Figure 61 Accessing unpublished pages from the *File* menu

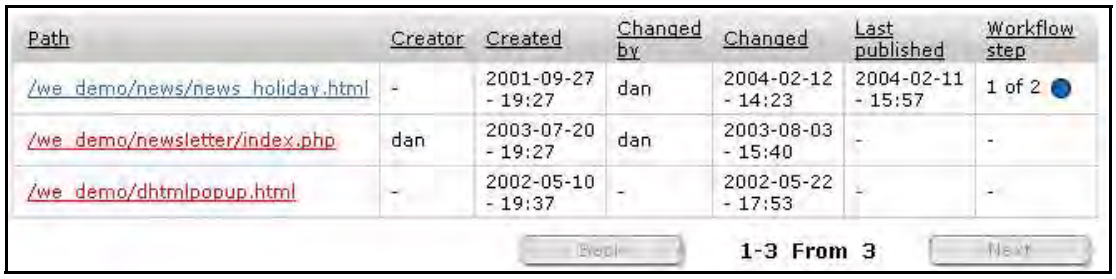


On this page you can view a table with all of the unpublished pages or objects that exist in the system (see Figure 62, "Table of unpublished pages" on page 64). The following information is displayed about the pages or objects:

- *Path*. Where in the system is the document/object?
- *Author*. Who is the creator of the document/object?
- *Created*. When was the document /object created?
- *Changed by*. Who had modified the document /object in the course of the workflow?
- *Changed*. When was the change made?
- *Last published*. When was the document /object last published?
- *Workflow step*. Where in the workflow is the document /object currently found? (This column exists only if the Workflow Module is installed.)

  If you click on a criterion (that is, a value found underlined in the header row of the table), the table will sort the output according to the selected criterion. If you click on the actual path of a document /object, the system will take you directly to the unpublished document /object.

Figure 62 Table of unpublished pages



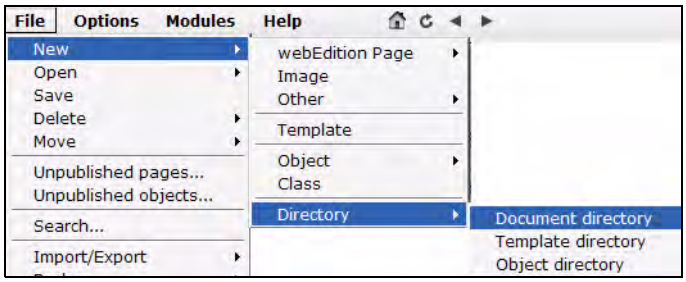
3.2.6 Modifying and updating a document

If you want to modify an existing document, select *File* > *Open* > *Document* from the *File* menu on the webEdition main screen, or single-click the icon in the explorer menu that represents the document you wish to change. (If you double-click an item, a new browser window opens, displaying the page.) Go to the *Edit* view and enter the new/updated text. You can view the changes in the *Preview* view. If everything is correct, save (and/or publish) the document.

3.2.7 Creating a new document directory

You can create a new directory within webEdition by selecting *File* > *New* > *Directory* > *Document directory* (see Figure 63). The new directory will be saved in the database and will therefore be displayed in the webEdition explorer menu. Any directory created on the server without using webEdition will not be displayed in the webEdition explorer menu. For more information about the *Browse Server* feature, see page 41.

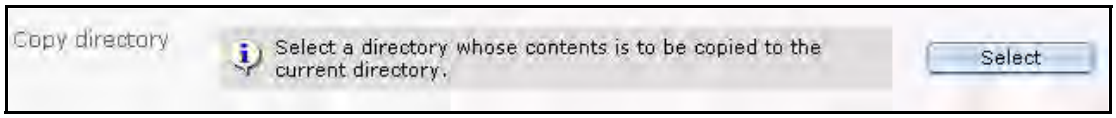
Figure 63 Creating a document directory



3.2.8 Copying the contents of one directory to another

All the contents of a webEdition directory (including sub directories) can be copied to your currently active directory in one step. While in the *Documents* view, if you click on a folder in the explorer menu, the Document Directory *Properties* view opens (see Figure 64):

Figure 64 Directory *Properties* view



In the *Copy directory* area, click on *Select.* A browser opens from which you can choose the directory whose contents are to be copied to the current one.

3.2.9 Rebuilding static and dynamic Web pages

After making substantial changes to a static Web page, you must synchronize the content of your database with the content of the directories on your Web server. When you use the rebuild feature, the new contents or new features in the webEdition interface will be regenerated and then made visible to visitors to your site You can regenerate either your entire Web page or only sections of it.

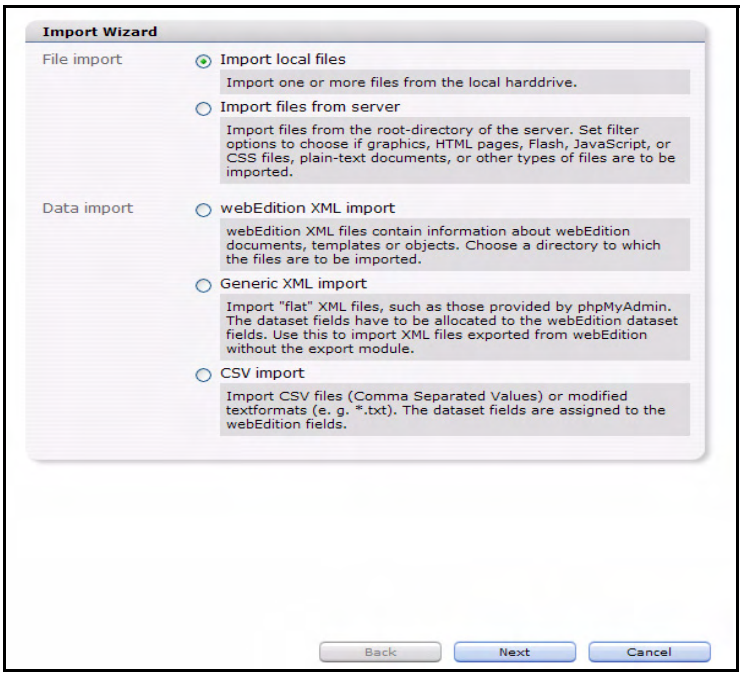
You can rebuild web pages using the *File > Rebuild...* command. Detailed instructions regarding how to use this feature are found in *The webEdition Installation, Update and Backup Procedures*.

3.3 Using the Import feature

webEdition allows you to import different types of files and data, and it offers an Import Wizard to help you with such actions (see Figure 65). The Import Wizard has two parts:

- *File import*. You can import files from two locations:
  - — from your local hard drive
  - — from your server
- *Data import*. You can import the following data files
  - — generic XML files
  - — CSV files

Figure 65 The Import Wizard



The following sub sections descibe procedures for importing files into webEdition.

3.3.1 Importing local files

Procedure 2 Importing local files

In the webEdition main screen

1 Open the Import Wizard by clicking *File* > *Import/Export* > *Import*.

The Import Wizard opens.

2 Activate the *Import local files* radio button.

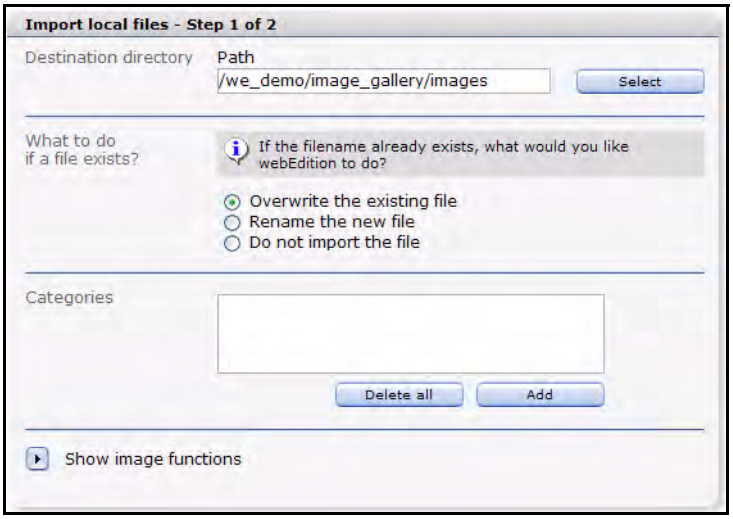
3 Click the *Next* button.

The Import local files window opens (see Figure 66).

4 Choose a destination directory by clicking the *Select* button adjacent to the *Path* field.

5 Instruct webEdition on what to do if the file name exists, then click the *Next* button. You can also add the files to a previously defined *category* and define rules for uploaded images in the *image functions*

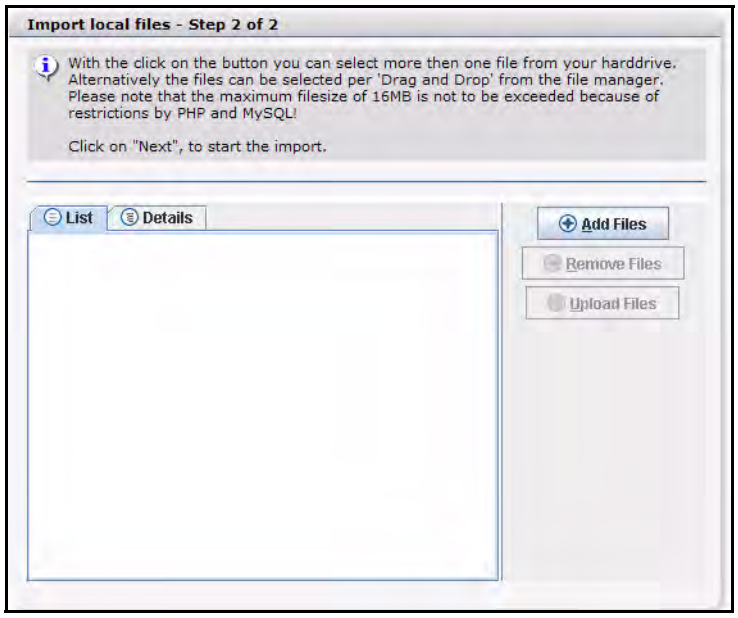
Figure 66 Importing local files



6 Click *Next*

The Java Applet starts and the Upload window opens - you might have to allow the execution of the applet in your browser!

Figure 67 Importieren Bildschirm

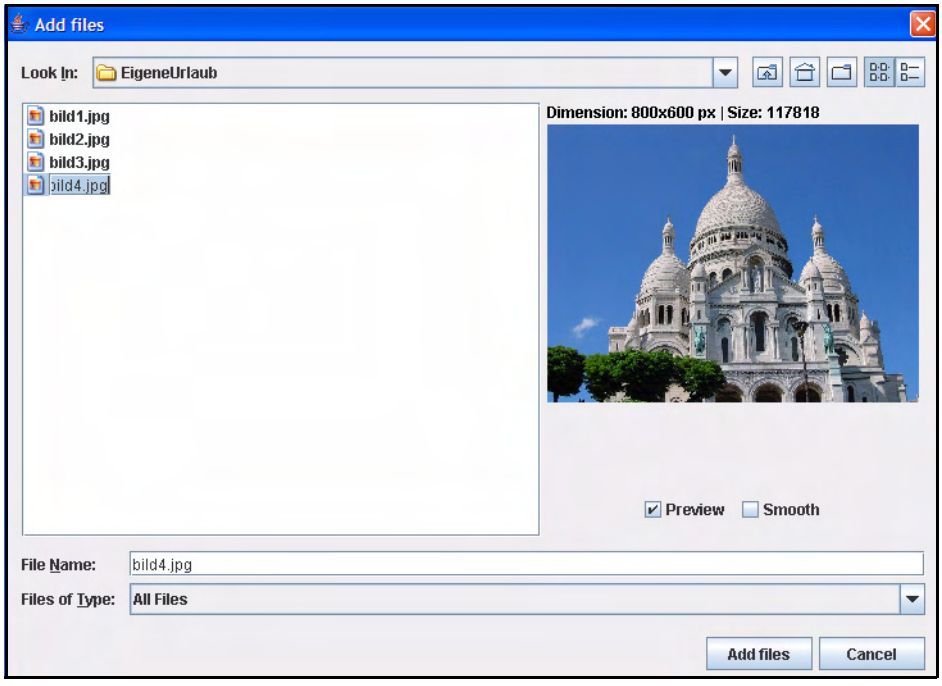


7 Click on *Add files*

The ADD FILES window is opened

8 Navigate to the folder and select the files to be imported

Figure 68 Dateien hinzufügen

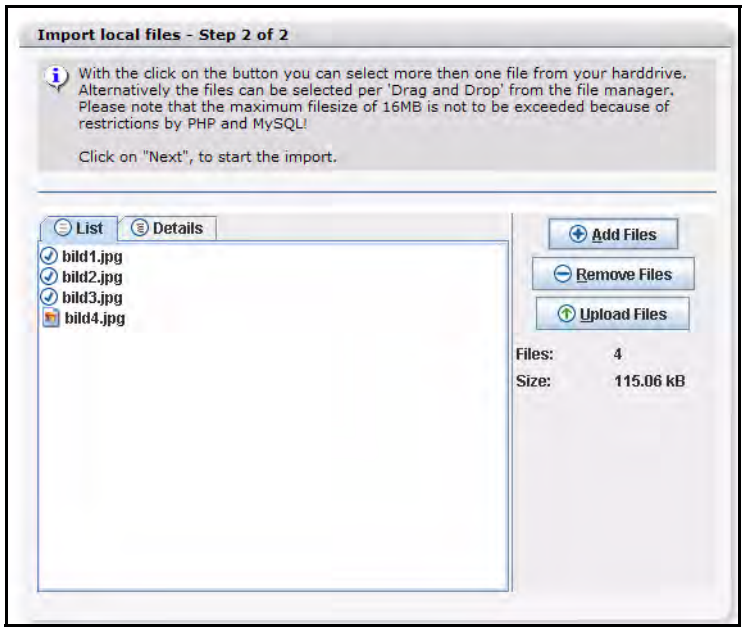


Note: You may select multiple files with the *Ctrl* or *Shift* key (Windows) pressed

9 After finishing your selection, click *Add files*

The selected files are displayed as a list - click on Details to display further information about the files

Figure 69 List of selected files



10 Click *Upload Files*

The files are now uploaded; after success the following message is displayed:

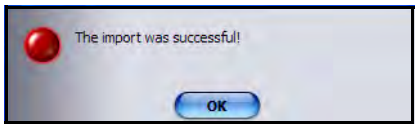


Figure 70 Import successful

11 You may now add more files or start working in webEditon with the imported files.

Note: You may also add the files via *Drag&Drop*!

12 Click *OK*.

You have completed this procedure

3.3.2 Importing files from your server

Procedure 3 Importing files from your server

In the webEdition main screen

1 Open the Import Wizard by clicking *File* > *Import/Export* > *Import*.

The Import Wizard opens.

2 Activate the *Import files from server* radio button.

3 Click the *Next* button.

The Site Import window opens (see Figure 71).

4 In the *Directories* area, choose a source and destination directory by clicking the *Select* buttons.

5 Use the checkbox in the *Import* area to indicate the type of file you wish to import.

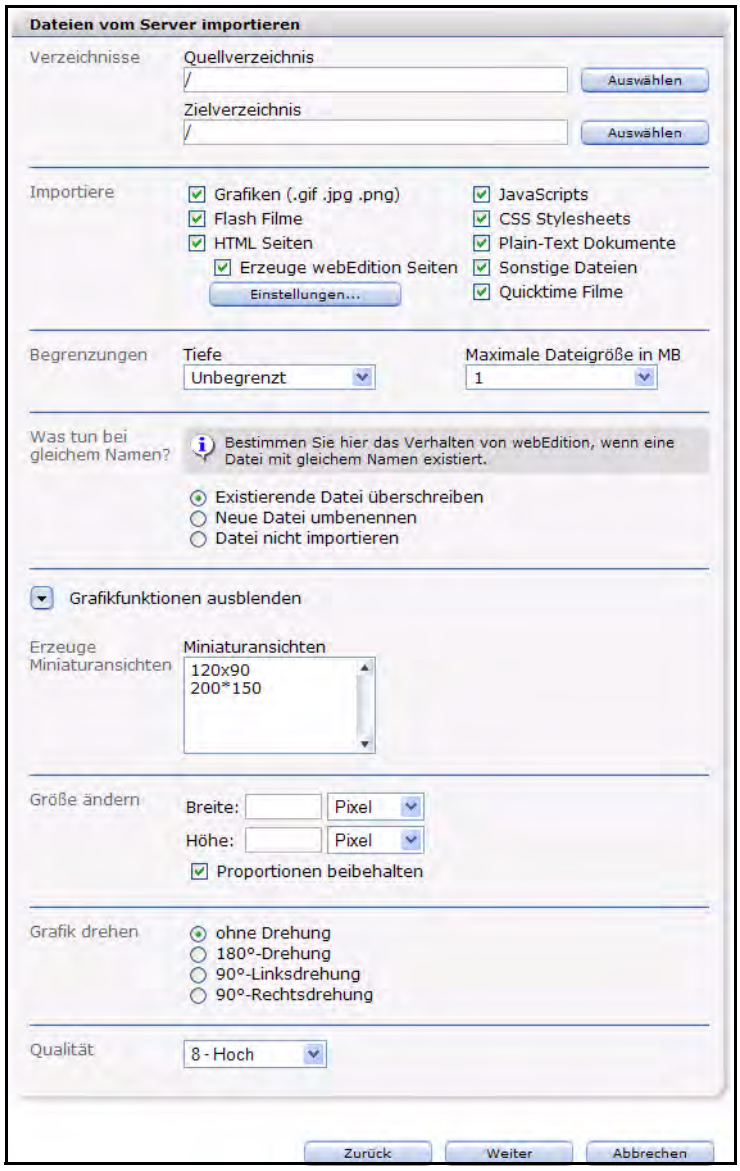
6 From the two select boxes in the *Limits* area, choose the depth and maximum file size for the import.

The *Depth* select box allows you choose the number of recursive directories that are to be imported. For example, if you have the following directory structure /news/images/gifs, and you you import files from the *news* directory with *depth = 0*, webEdition will only import files from the *news* directory. Files from "images" will not be imported. If you set the depth to

depth = 1, webEdition will also copy files from the *images* directory. If you set the depth to *depth = 2*, files from the *gifs* directory will be imported.

7 Instruct webEdition on what to do if the file name exists, then click the *Next* button.

Figure 71 Importing files from a server



A progress bar appears at the bottom of the screen to tell you that upload is in progress. When the import has completed, the system will tell you the outcome of the action. Click OK.

8 You have completed this procedure.

3.3.3 Importing generic XML data

Procedure 4 Importing generic XML data

In the webEdition main screen

1 Open the Import Wizard by clicking *File* > *Import/Export* > *Import*.

The Import Wizard opens.

2 Activate the *Generic XML Import* radio button.

3 Click the *Next* button.

The Generic XML Import window opens (see Figure 72).

4 In the *importing* area, choose a source file by clicking the *Select* button.

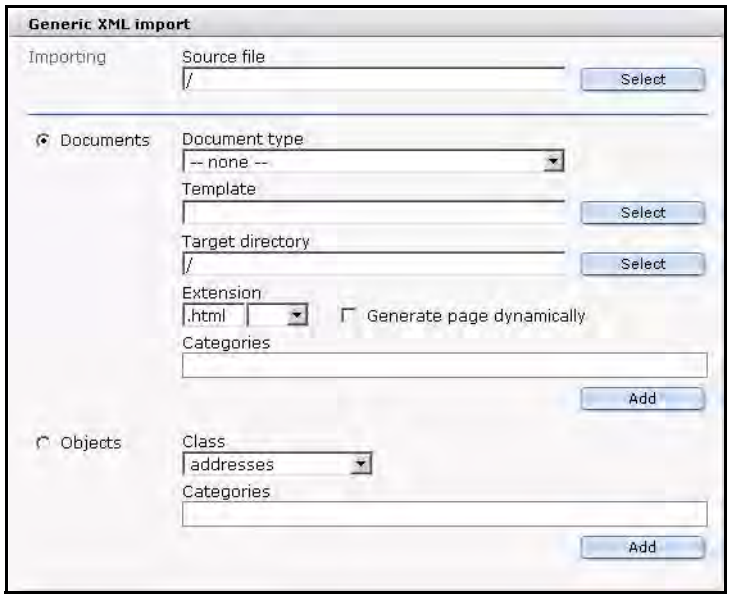
5 Use the radio buttons to select whether the import is a document or object.

a If you choose *documents*, select the document type, template, target directory, extension and category for the import.

b If you choose *objects*, select the object class and category for the import.

6 Click the *Next* button.

Figure 72 Generic XML Import



A progress bar appears at the bottom of the screen to tell you that upload is in progress. When the import has completed, the system will tell you the outcome of the action. Click OK.

7 You have completed this procedure.

3.3.4 Importing CSV files

Procedure 5 Importing CSV files

In the webEdition main screen

1 Open the Import Wizard by clicking *File* > *Import/Export* > *Import*.

The Import Wizard opens.

2 Activate the *CSV import* radio button.

3 Click the *Next* button.

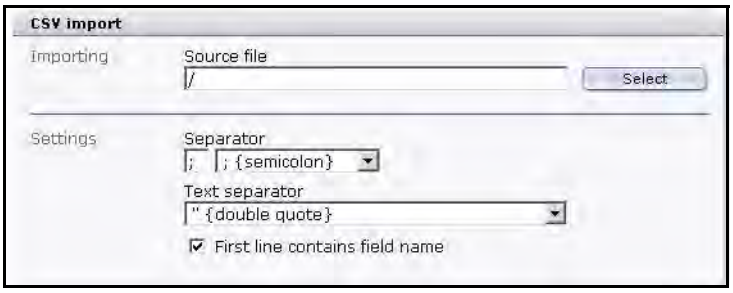
The CSV Import window opens (see Figure 73).

4 In the *importing* area, choose a source file by clicking the *Select* button.

5 In the *Settings* area, choose the type of separator and text separators used in your document, and indicate if the first line in the file uses a fieldname.

6 Click the *Next* button.

Figure 73 CSV Import



A progress bar appears at the bottom of the screen to tell you that upload is in progress. When the import has completed, the system will tell you the outcome of the action. Click OK.

7 You have completed this procedure.

3.3.5 webEdition XML import

This function serves exclusively the import of XML files which were generated with the webEdition Export module.

Procedure 6 webEdition XML import

In the webEdition main screen

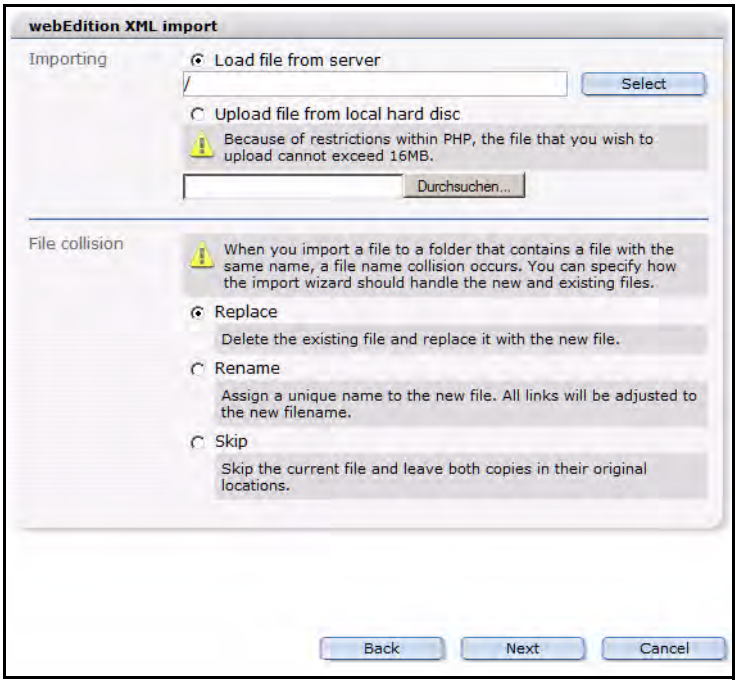
1 Open the Import Wizard with *File* > *Import / Export* > *Import*

2 Activate *webEdition XML Import* radiobutton

3 Click on *Next*

The webEdition XML Import window opens (see Figure 74)

Figure 74 webEdition XML Import



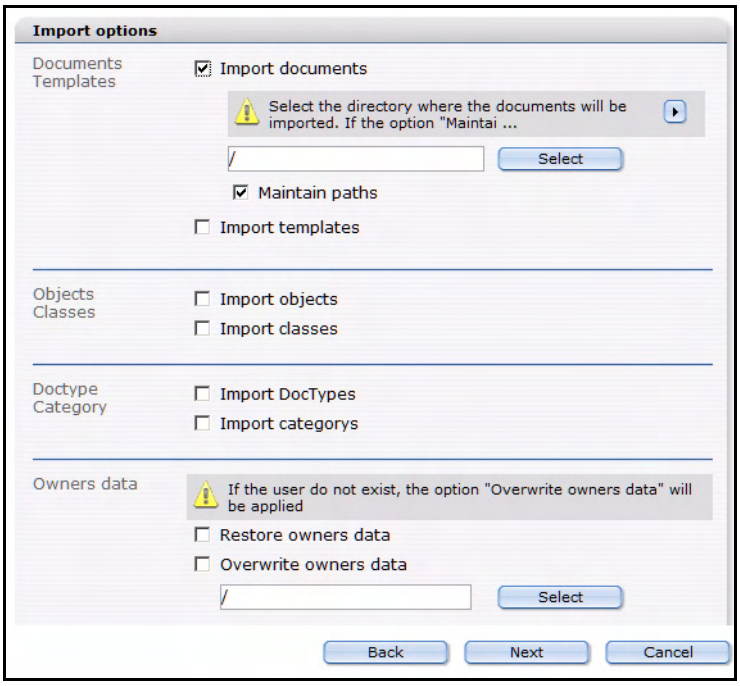
4 Select in the *Importing* area the file to be imported. It can be located on the server, as well as on your local hard disk

5 In the *File collision* area you determine how the system should behave, if already files of the same name should exist.

6 After you have dripped your choice, click on the *Next* button.

The Import options window opens (see Figure 75)

Figure 75 webEdition XML import options



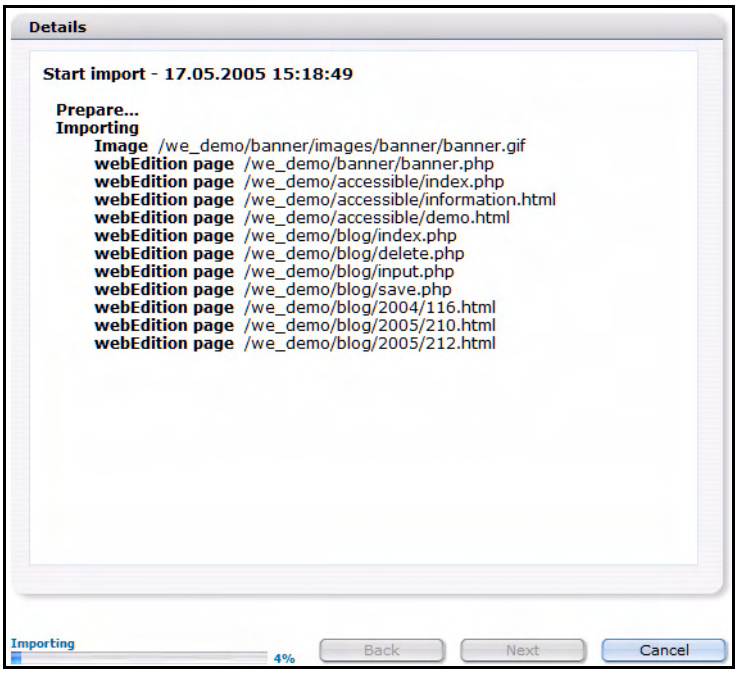
7 In this window are, depending on your installed modules and the data contained in the exported file, several areas:

- *Documents / templates*: Activate these checkboxes, so the documents and/or templates from the file to be imported are imported. With activated check box you can determine with the *Select* button a destination directory
- *Objects / classes*: Activate these check boxes to import objects and/or classes. This area is only displayed if the DB/OBJECT module is installed and objects or classes exist in the file to be imported.
- *Doctype / category*: These check boxes must be activated if you want to import document types and/or categories
- *Owners data*: Here you find the check boxes *Restore owner's data* and *Overwrite owner's data*. This area is only displayed if the user administration (PRO) is installed and user's data were coaccepted in the export.
  - — Activate the check box *Restore owner's data*, around the user to whom the data "belonged" originally to assign the rights also on the target system. If the data User1 belonged, is also on the target system User1 of the owners. The user User1 must exist for it of course on the target system. Should this not be the case, the files are imported general-accessibly. This also applies to several users: If the data belongs to User1, User2 and User3, these rights on the target system are recovered, provided that these users also exist there.
  - — Activate the check box *Overwrite owner's data* and select a new user, while you click on the *Select* button to associate a new owner to the imported files. If both check boxes are activated, the rights are associated in the imported data to the same-being users, provided that these exist. Should the users not exist in the webEdition instance in which is imported, the rights go over to the user selected under *Overwrite owner's data*.

Click on *Next*

The details window opens (see

Figure 76 Import details



8 Click after successful end of the import, whose progress you can read in the beam at the bottom on the left, on the *Cancel* button

Herewith the import of the webEdition XML data is concluded successfully

3.4 Working with graphical documents

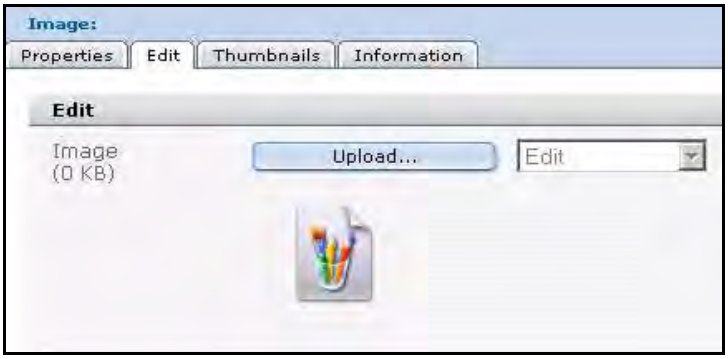
webEdition allows you to access not only webEdition pages, but also other graphical documents, which you can select from the *File* > *New* > *Image* drop-down menu.

webEdition differentiates between layout images, which are stored centrally on the host server in an image directory, and the images that vary depending on the accompanying content. The latter images are used by the editor and have to be inserted in the pages. Content images have to be uploaded before importing them to the webEdition database. If you wish, you can also import all the layout images to the database.

3.4.1 Uploading an image

Click *File* > *New* > *Image*. The *Upload image* dialog box opens in the *Edit* view (see Figure 77).

Figure 77 Uploading an image

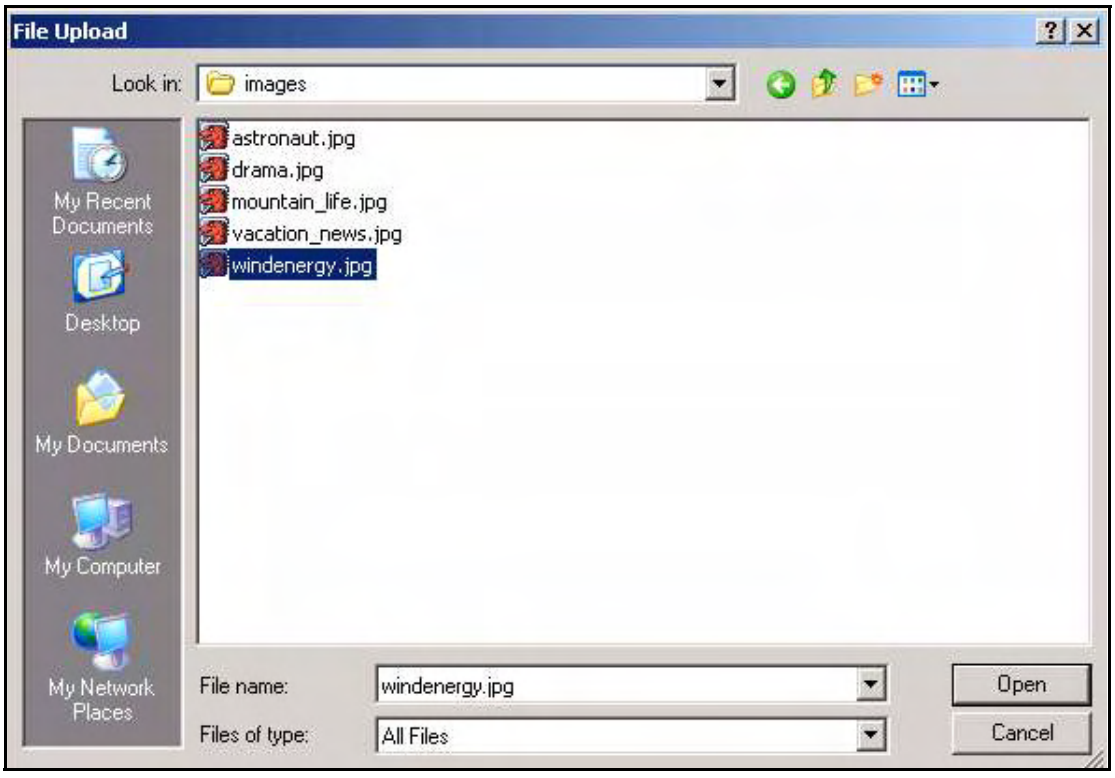


If you wish to upload an image from your hard drive to the webEdition database, click on the *Upload* button. A window appears in which you can browse your hard drive (see Figure 78).

Figure 78 Browsing your files for an image

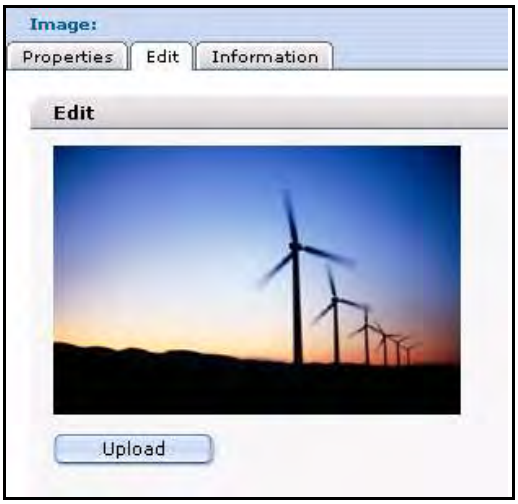
Click *Browse*. You will see the select window, in which you can browse for an image on your computer (see Figure 79).

Figure 79 Selecting an image to upload



After you have found the image that you want, double-click it and it will appear in the search window. Click *Upload* in this window to display the selected image in the *Edit* view (see Figure 80).

Figure 80 Viewing an image in the *Edit* view



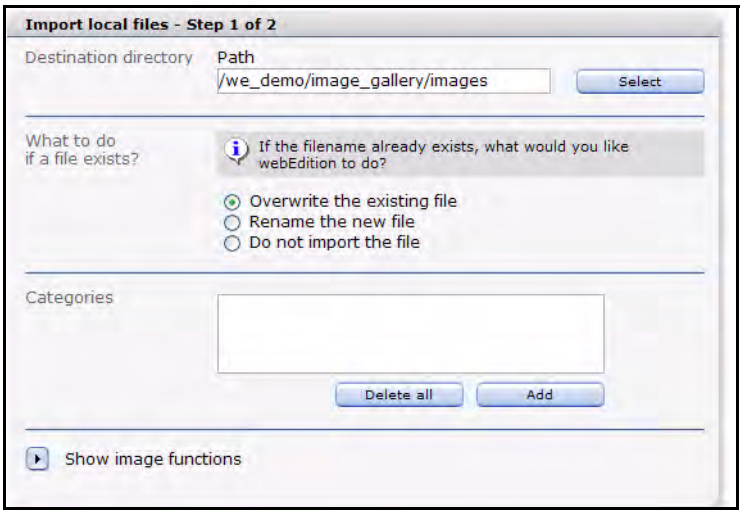
Use the *Properties* view to enter the name and directory in which you want to save the file. If you do not enter a new name, the image will be saved with its original name (default setting). If you clicked *Save*, you will be able to view the uploaded image in the directory you selected in the explorer menu and will be able to include it in webEdition pages. To learn how this is done, please consult Section 3.7.9, "Image field (<we:img>)" on page 106.

3.4.2 Importing more than one image

In webEdition you have the option of importing several images at once. Choose the item *File* > *Import/Export* > *Import...*. An import file dialog box opens (see Figure 65, "The Import Wizard" on page 65).

Activate the *Import local files* radio button, and click *Next*. The following *Import local files* screen opens (see Figure 81):

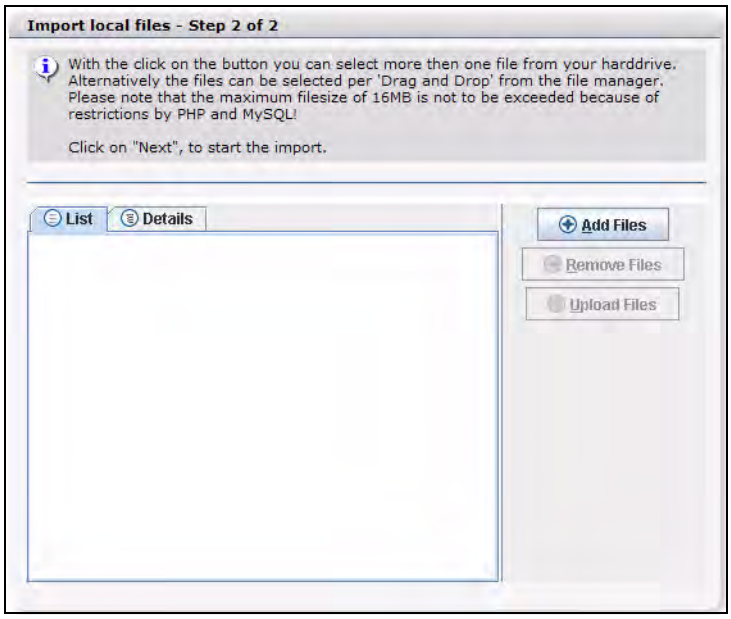
Figure 81 *Import local files* - step 1 of 2



Click on the *Select* button in the *Destination directory* area. The webEdition file manager opens, allowing you to choose the directory to which the images are to be imported. In this file manager, all directories within the root directory (/htdocs) of the

Web server are displayed, therefore directories that are not managed with webEdition are also shown. The following window appears (see Figure 82):

Figure 82 Import local files - step 2 of 2

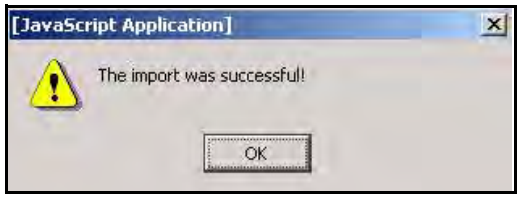


webEdition asks you *What to do if a file exists?* in the lower area of the window. You must choose from one of three options. If, for example, you have modified an image and you wish to replace the old image, activate the *Overwrite the existing file* radio button to replace the old image. If the filename already exists in the target directory and you wish to rename the new file select *Rename the new file*. webEdition renames files by adding a number to the file name (for example, bg.gif becomes bg_0.gif). You can also instruct webEdition not to import a file if its file name already exists.

Click the *Next* button. Click the *Browse...* button to find an image on your local drive. You can repeat this action to import additional files.

Click *Next* to start the import. When the process is complete, the following message appears (see Figure 83):

Figure 83 Import confirmation screen



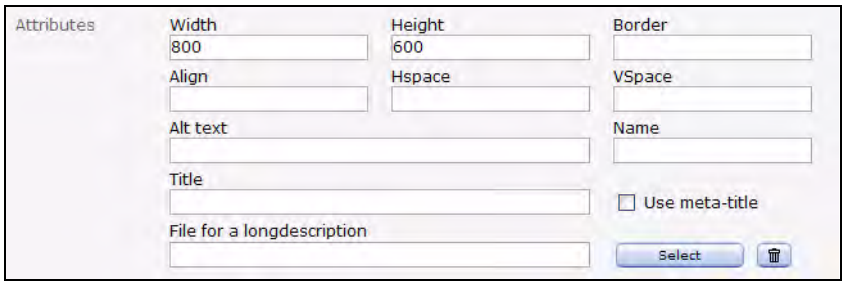
Click *OK* to complete the procedure.

3.4.3 The image *Properties* view

The *Properties* view allows you to define the display parameters for an image. There are three areas, *Path*, *Attributes*, and *Hyperlink*.

- *Path* area. The *Path* area allows you to give (or change) a file name, select a directory, or choose a file extension for your graphic.
- *Attributes* area. Use this area to modify the appearance of your image file (see Figure 84).

Figure 84 The *Properties* view, *Attributes* area



The default settings are the values of the unchanged, original image.

— *Width*. Image width in pixels.

— *Height*. Image height in pixels.

— *Border*. You can have the browser draw a border around the image. The entry is calculated in pixels and specifies how thick the border will be

— *Align*. Entering left or right will cause the browser to align the image left or right. Accompanying text on the Web page is displayed to the left or right of the image and above and below it.

— *Hspace*. This field allows you to specify how far (to its right and left) the image is to be from other elements.

— *Vspace*. This field allows you to specify how far the image is to be from other elements above and below it.

— *Alt Text*. This is the text that is displayed while the image is loading.

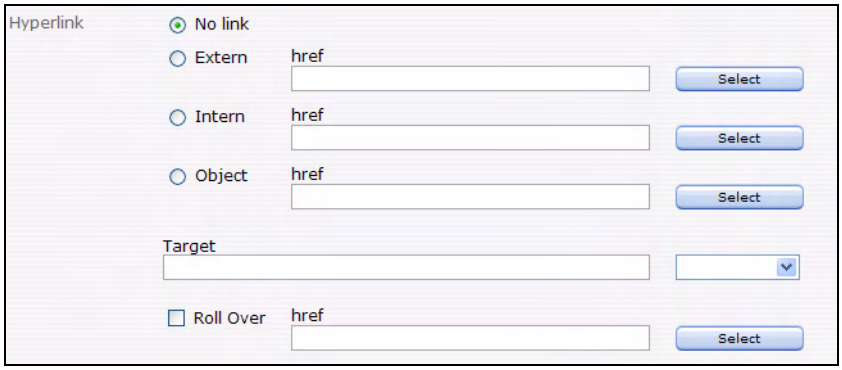
— *Name*. Image name.

— *Title*. Enter a title for the image. Check *USe meta-title* to use the *Title* used in the *Meta tags* area

— *File for a longdescription*. Select a file for a long description of the image; e.g. for the description of a chart for a partially sighted person

- *Hyperlink*. In this area the image can be linked to a URL (see Figure 85):

Figure 85 The *Properties* view, *Hyperlink* area



— *No link*. Image is not linked.

— *Extern*. External link, e.g. www.webEdition.de, can be selected using the select button.

— *Intern*. Internal link, can be selected using the select button.

— *Target*: Target denotes the location where the linked Web page should appear:

  – empty field, Web page will open in the same window or frame

  – _blank, a new browser window will open

  – _top, the page will be loaded in the same window on the uppermost level, even if it has been called up from a frame.

  – or you can specify the exact name of the frame or window in which the Web page will be displayed.

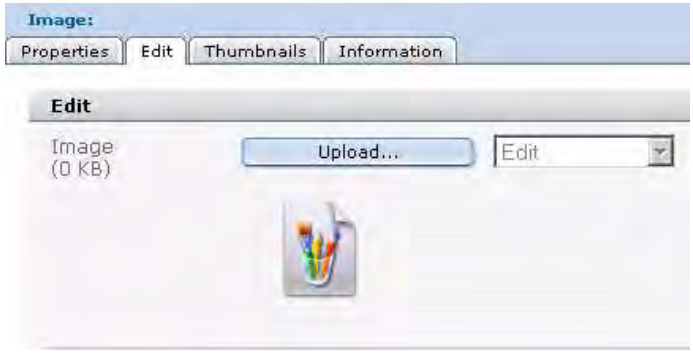
— *Roll Over*. If this checkbox is activated, you can choose a graphic using the select button which will appear to replace an image on a Web page as soon as one moves the mouse pointer over the original image. (This Web page feature is commonly called a roll-over or mouse-over effect)

3.4.4 The image *Edit* view

The image *Edit* view allows you to upload a new image from your local hard disk, or to overwrite an existing image. Click *File* > *New* > *Image* to open the *Image Edit* view (see Figure 87).

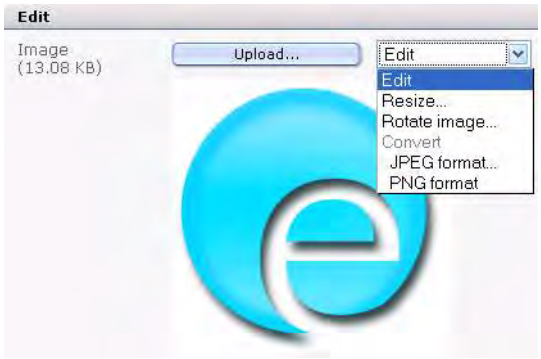
To select a file, click the *Upload* button, then click *Browse* to choose your file. (See also Figure 78, "Browsing your files for an image" on page 74 and Figure 79, "Selecting an image to upload" on page 74.)

Figure 86 Uploading a new image



Once your new file has been uploaded, you can use the select box shown in Figure 87 to perform the following editing functions:

Figure 87 Editing an image

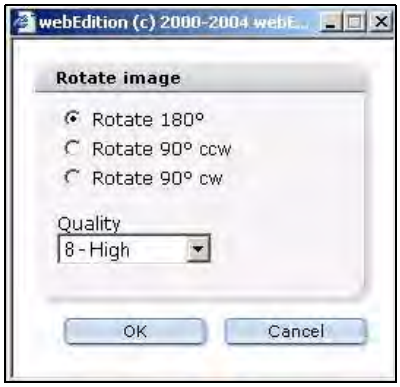


- *Resize*. Define the size of the image. If the image is a JPEG, you can also choose the kind of compression.

Figure 88 Resizing an image

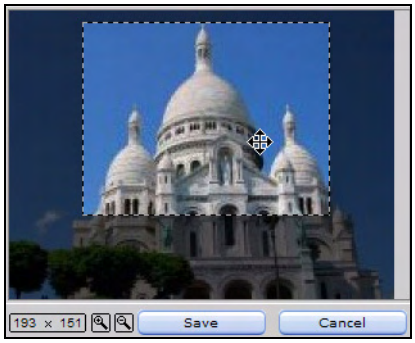
- *Rotate image*. You can rotate the image here. If the image is a JPEG, you can also choose the kind of compression.

Figure 89 Rotating an image



- *Crop image*. Use this command to crop the image. Simply drag a box with your mouse to use only that part of the image. The exact size in pixels is displayed at the bottom.

Figure 90 Cropping an image



- *Convert*. Use this command to change the format of the image. If the image is a JPEG, you can also choose the amount of file compression.

Figure 91 Converting and image

3.4.5 The images *Information* view

For a description of this view, see Section 3.1.3, "The Documents Information view" on page 56.

3.4.6 The images *Thumbnail*s view: thumbnail formatting

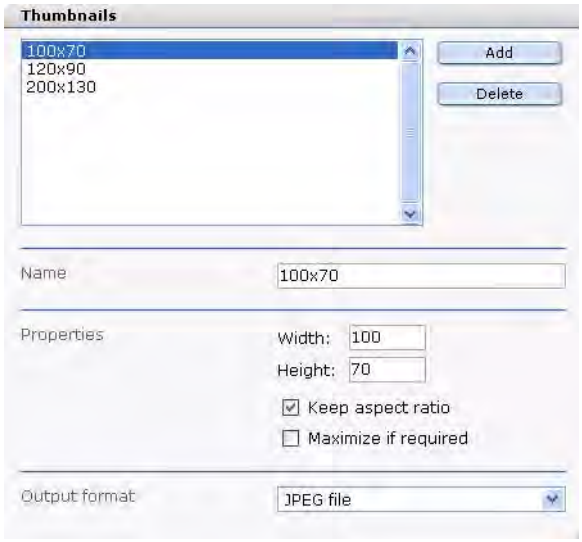
webEdition allows you to create thumbnails from your existing images so that they can be displayed on your Web site in a picture gallery.

Creating thumbnails expends considerable processing resources, which can lead to latency. webEdition does not create thumbnails from your original images in real time

each time your Web page is viewed by a visitor. To ensure that your thumbnails will load efficiently, you must pre-define the size and output format for your thumbnails using the *Options* > *Thumbnails* menu. By pre-defining your thumbnail formats, you can ensure that webEdition will automatically upload all of your thumbnails in the same size and format.You can pre-define, store and edit multiple thumbnail formats.

To create a new thumbnail format, click *Options* > *Thumbnails*. The *Thumbnails format* dialogue box opens. See Figure 92. Click the *Add* button. A dialog box opens and asks you to enter a name for your thumbnail format. Enter a unique and descriptive name and click *OK*. The name of your new format will appear in the text box on the *Thumbnails formats* dialog box.

Figure 92 Defining thumbnail formats



Below the text box shown in Figure 92 are the following areas that you can use to define the characteristics of each thumbnail format:

- *Name*. If you have defined more than one thumbnail format, the *Name* field indicates which format you are viewing. You can also use this field to change the name of an existing thumbnail format.
- *Properties*. Use the *Width* and *Height* fields to define the size of the thumbnails. If you activate *Keep aspect ratio*, the proportions of the image will be retained. If the aspect ratio of the original image is different than the defined aspect ratio, the width and/or the height of the created thumbnail will never be larger than one of the adjusted values.

  Example
  The original image has a width of 100 and a height of 80.

  The defined values for the thumbnail are 50 (width and height) and *Keep aspect ratio* => The thumbnail will be 50 x 40.

The defined values for the thumbnail are 50 (width), 20 (height) and *Keep aspect ratio* => The thumbnail will be 25 x 20.

If you indicate only one of the two values or if you indicate 0 as a value, the missing value (and/or the value 0) is computed proportionally.

Example
The original image has a width of 100 and a height of 80.

The defined values for the thumbnail are 50 (width) and 0 (height) => The thumbnail will be 50 x 40.

If you activate *Maximize if required*, the thumbnail will be scaled up if the original image is smaller than the defined values for the thumbnail.

If you activate *Maximize if required* and the original image is smaller than the thumbnail, the original images will be used as thumbnail.

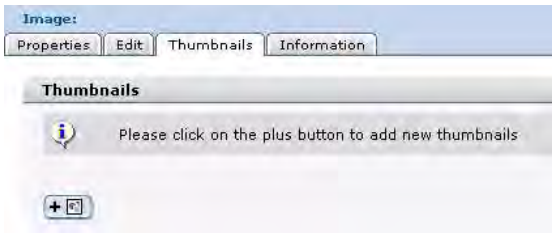
3.4.7 Associating an image with a thumbnail format

Click *File* > *New* > *Image*. The Image: Edit view opens.

Figure 93 Uploading a new image

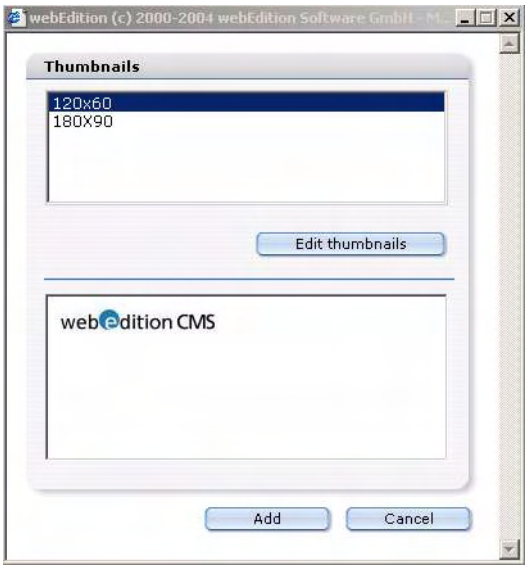
After uploading an image from your server or computer, click the *Thumbnails* tab. Click on the add image (the plus "+" icon).

Figure 94 Thumbnails tab



The formats that you pre-defined (see Section 3.4.6, "The images Thumbnails view: thumbnail formatting" on page 80) appear in a select box above the image you have uploaded. Select the format for the image and click *Add*.

Figure 95 Formatting a thumbnail



After formatting your image, it will appear in your *Thumbnails* list. (See Figure 96). Click the *Save* button to complete the formatting.

Figure 96 Thumbnails list

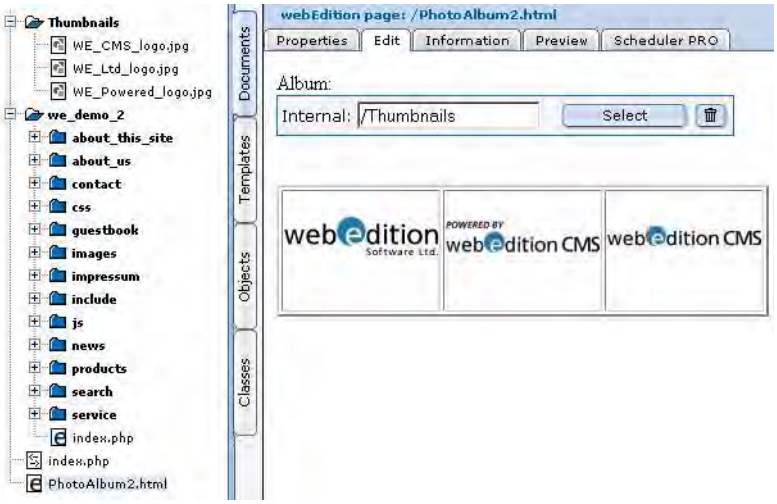
3.4.8 Displaying your photo album

Before you can upload images to a photo album, the template administrator must create a photo album template in which the thumbnails can be displayed. For details see Section 4.6.3, "Creating a photo album" on page 144.

Create a directory for your thumbnail image files. This directory will contain the images for your photo album. Move your image files into the designated directory (or directories). You can do so by clicking on your image and opening the *Properties* view. In the *Path* area, choose the directory for your image by clicking *Select* and browsing your directories.

Create a new document by clicking *File* > *New* > *webEdition Page* > *Empty page*. Give the file a name in the *Path* area. In the *Document* area, use the *Select* button to browse your system for the album template created by your template administrator. Select the template, then click *Save*. Click on the *Edit* tab. Use the *Album: Select* option to choose what photo album (directory) will be displayed (see Figure 97). You can preview your thumbnails by clicking the *Preview* tab. Save and publish your photo album.

Figure 97 Choosing a photo album directory



3.5 Working with non-graphical documents

webEdition allows you to work with a the following non-graphical document types, which you can select from the *File* > *New* > *Other* drop-down menu:

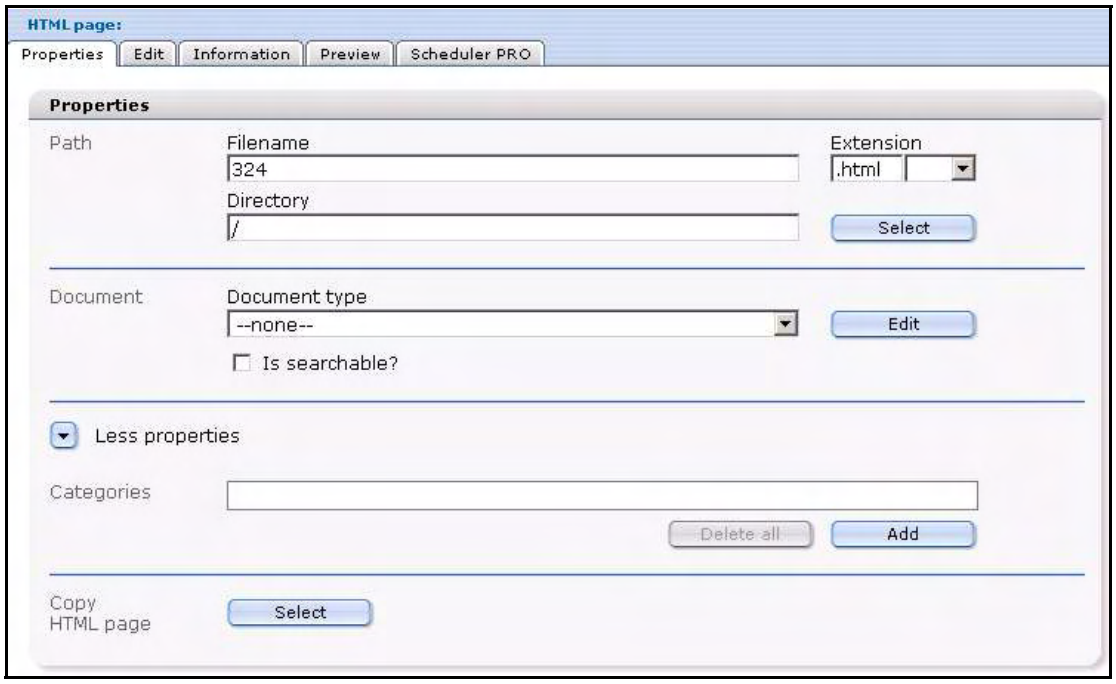
- HTML Page
- Flash movie
- Quicktime movie
- JavaScript
- CSS Stylesheet
- Other Files (*.doc, *.pdf, *.zip, *.txt)

3.5.1 Editing HTML pages

3.5.1.1 HTML document *Properties* view

You can use the HTML *Properties* view to program your own HTML pages (see Figure 98). You cannot integrate webEdition features into this page. However, document types and categories can be selected in order to include these pages in summaries or in the "Find" feature. To make this possible, you must activate the *Is searchable?* check box. Since simple HTML pages do not have fields for a title or a description, you can have summary pages or search results pages display the path for the page or its content (either in its entirety or logically reduced to a certain number of characters) instead of displaying a field.

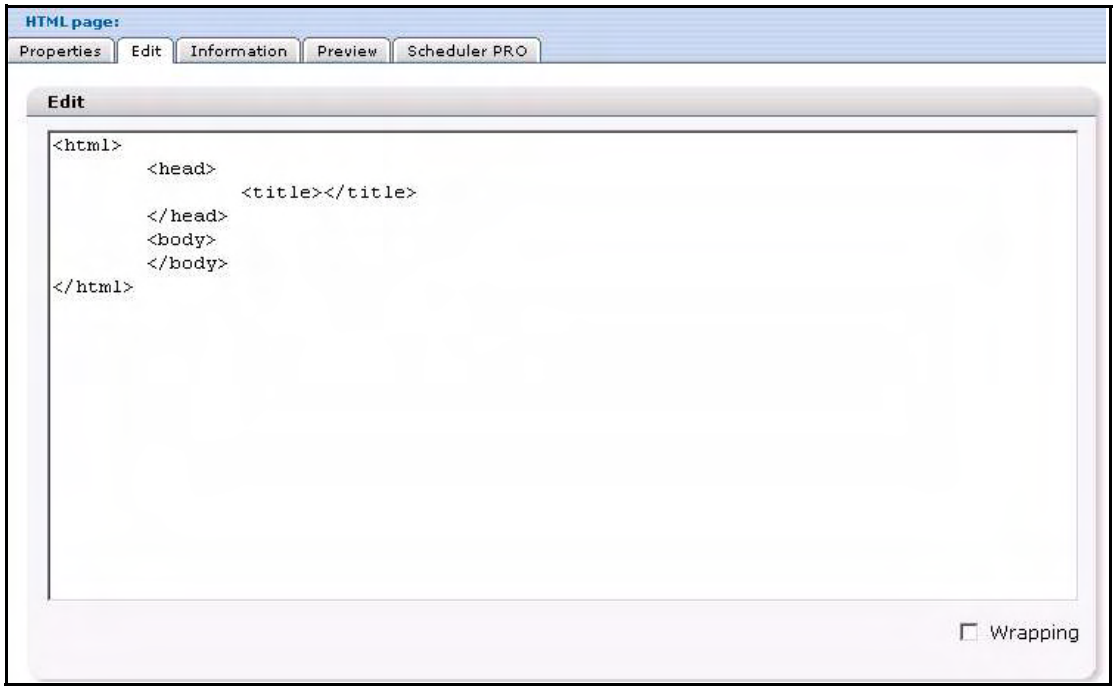
Figure 98 HTML *Properties* view



3.5.1.2 HTML document *Edit* view

You can enter HTML code in the *Edit* view (see Figure 99). In this case, webEdition acts as a simple text editing program. You can also program PHP scripts, Active Server Pages (.asp) or Java Server Pages (.jsp) in this view. To do this, you also have to select the corresponding extension when setting the properties for the page.

Figure 99 Editing an HTML page



3.5.2 Using Flash movies

3.5.2.1 Flash movie *Edit* view

You can use the *Edit* view to upload Flash movies. Follow the same procedure for uploading Flash movies that you followed when uploading images (see Section 3.4.1, "Uploading an image" on page 73).

Note: You cannot allocate a hyperlink to a Flash movie because the attributes for Flash movies differ from those for images.

3.5.2.2 Flash movie *Properties* view

The Properties view has the following areas (see Figure 100, "Flash movie Edit view" on page 87):

- *Path* area. The file name of the original movie is used as the default name.
- *Attributes* area. Several of the attributes are the same as those for images (). Additional attributes are:
  - — *Scale*:
    - With the default setting, (Omit Attributes or Show All) the complete movie will be displayed in the area specified. Furthermore, the aspect ratio for the movie is also retained. This means that the film will not appear distorted. Occasionally, bars will also appear on either side of the movie.
    - *No Border*. scales the film to fit the entire area specified while still maintaining the film's original aspect ratio. The film will not appear distorted. Sections of the film might occasionally be cut off.
    - *Exact fit*. displays the film exactly within the area specified. However, this setting doesn't retain the film's original aspect ratio, which might cause the film to appear distorted.
  - — *Play*. If you wish the file to begin playback immediately, do not set this attribute. However, if you do not wish the file to begin playback until the user performs an action, set this attribute. This is equivalent to a STOP action in the first frame of your movie.
  - — *Quality*. Here you can specify the quality of your movie.
  - — *Backgroundcolor*. This is the background color of your movie. Flash doesn't offer many options, but you can manually set the background color in this field.
  - — *Align*: Left, Right, Top, Bottom.

— *Scaled Align*:

– Left, Right, Top, Bottom aligns movies with the edge of the browser window you specify and, if necessary, cuts off the edges of the remaining three sides.

– Top left, Top right, Bottom left, Bottom right aligns movies with the corner of the browser window you specify and, if necessary, cuts off the edges of the remaining two sides.

— *Loop*. Replays the Flash movie once it finishes. Select *Yes* or *No*.

- *Other* area

— *Pluginspage*.

— *Codebase*.

Figure 100 Flash movie *Edit* view

3.5.3 Using Quicktime movies

Follow the same procedure for Flash movies. Most attributes are the same with the exception of the following:

- *Scale*.
- *Autoplay*.
- *Show Controller*.
- *Volume*.
- *Hidden*.

See the vendor documentation for more details on Quicktime movie attributes.

3.5.4 Editing JavaScripts

3.5.4.1 JavaScript *Properties* view

You can use webEdition to create or edit JavaScripts. If you click on a JavaScript (.js) document or you select *File* > *New* > *Other* > *Javascript*, the Java script *Properties* view opens.

If you wish to copy an existing JavaScript from your directories, click the *Select* button in the *Copy Javascript* area. Thereafter you can rename the file by using the *Filename* field and then saving the document.

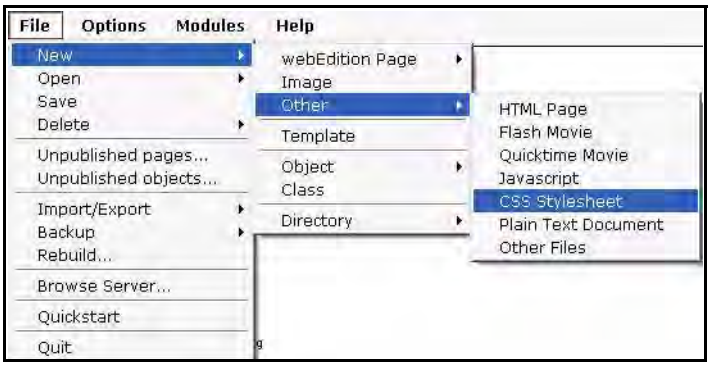
3.5.4.2 JavaScript *Edit* view

You can write your own JavaScripts (or modify an existing one) in the *Edit* view. Thereafter you can include your JavaScript in a template by using the <we:js> tag. For example: `<we:js id="34"/>`. This code includes the JavaScript with the ID 34 in the template. The views are the same as those shown in Section 3.5.5, "Creating a cascading style sheet (CSS)" on page 88.

3.5.5 Creating a cascading style sheet (CSS)

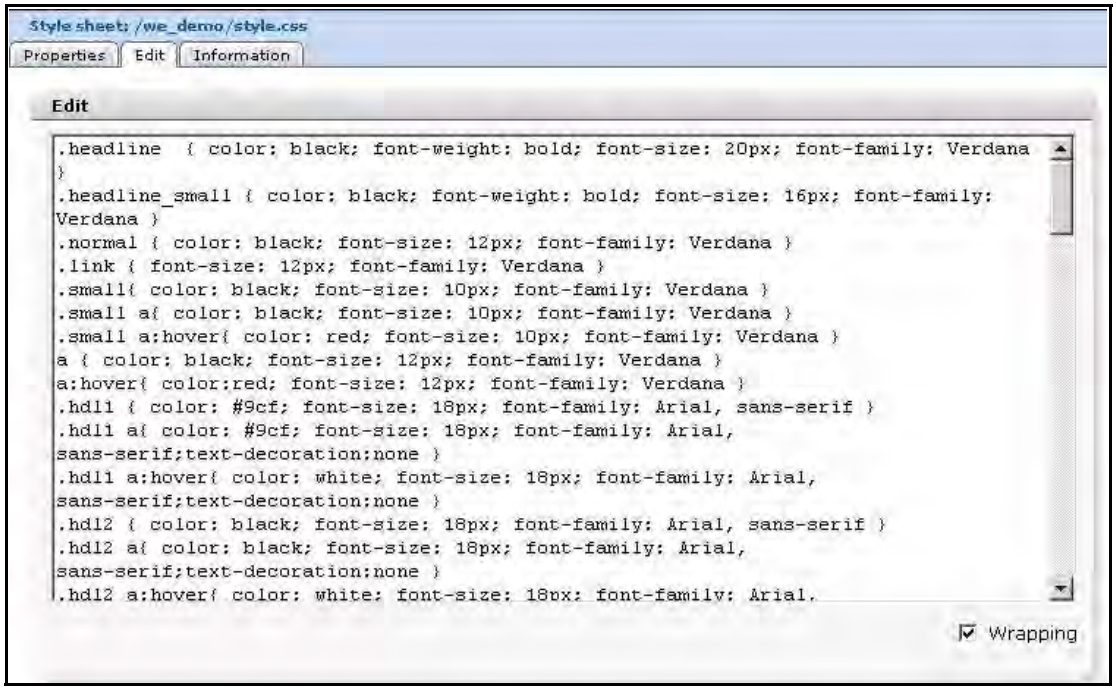
You can use webEdition to create or edit a CSS. To do so click *File* > *New* > *Other* > *CSS Stylesheet*.

Figure 101 Creating a CSS



You can include a style sheet in the template by using the <we:css> tag. For example: `<we:css id="89"/>`. This code includes the style sheet with the ID 89 in the template.

Figure 102 Cascading style sheet in the *Edit* view



3.5.6 Other files

The following additional documents can be imported into webEdition so that they are available to be downloaded by visitors to your Web site:

- Microsoft Office documents (.doc, .xls, .ppt)
- PDF documents (.pdf)
- Zip files (.zip)
- Stuffit files (.sit)
- Programs (.exe)
- Mac Binhex files (.hqx)
- BIN files (.bin)

3.5.6.1 Uploading other files in the *Edit* view

To import a file, select the menu item *File > New > Other > Other Files*. The *Edit* view opens. Then follow the same procedure as you would when uploading an image (see Section 3.4.1, "Uploading an image" on page 73). The file name of the original file is used as the default name.

3.5.6.2 Other files *Properties* view

Select the directory in which you wish the document to be saved in the *Path* area of the *Property* view.

To make the file available to visitors to your site, link the document to a text entry or to a link list.

> ***Note:*** The size of the upload can be limited. The default settings for MySQL are such that you can save up to a maximum of 1 MB of data per SQL-instruction. For PHP the default setting for the maximum upload capacity is set to 8 MB. If you wish to upload larger files, change the settings in your MySQL and PHP configuration files, or speak with your service provider about increasing these default values.

3.6 Features in the *Options* menu

For details about the items available in the *Options* menu, see Figure 26, "The Options menu" on page 41.

3.6.1 Document types

The *Document types* feature is accessed from the *Options* menu. The *Document types* feature allows you to group documents with similar characteristics, or to assign characteristics to a new document that are similar to existing documents. Document types are intended to ease your workload by enabling you to set collective default attributes for documents. When you create a new page you will be asked to assign the document to a document type. When you do so, the new document will inherit the default values of the chosen document type.

The default values that you set will pre-define the location, template, and file extension. Once these values have been set as defaults, you will not need to set them again whenever you create a new webEdition page and assign it to a document type.

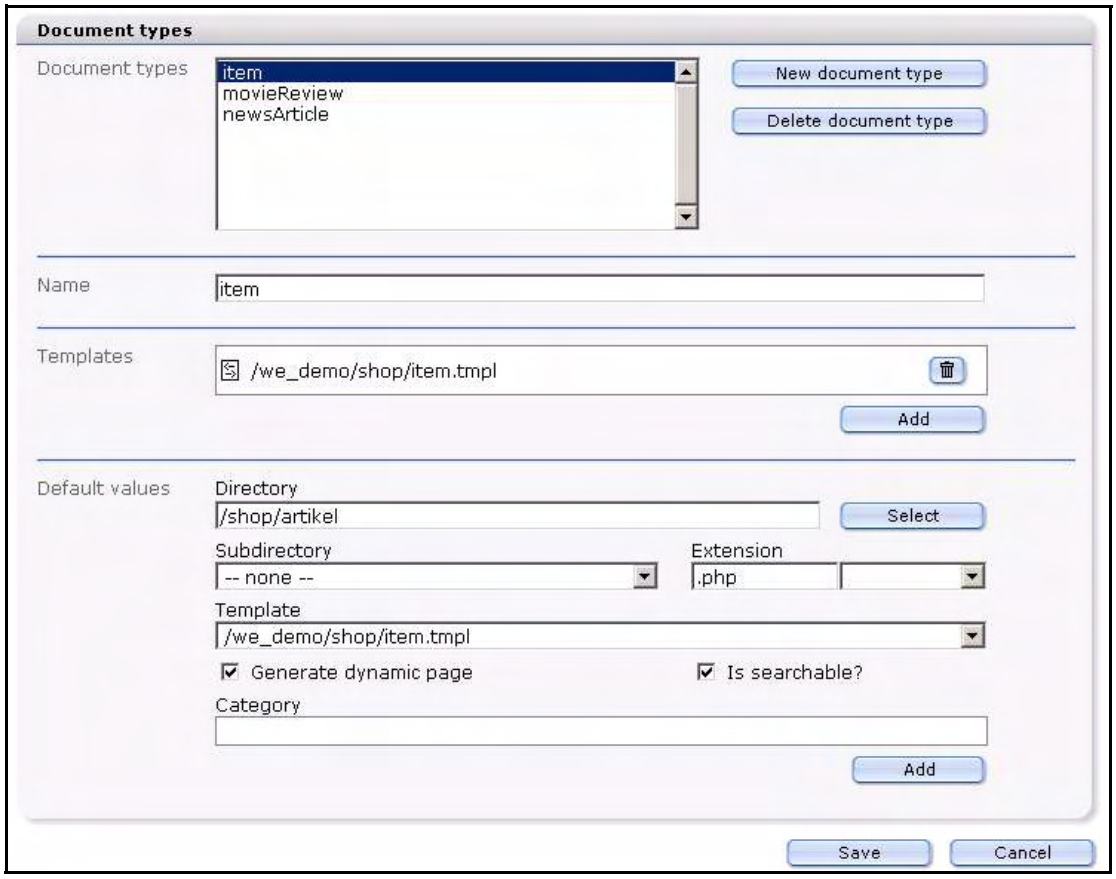
Secondly, because a document type can be allocated to any number of webEdition pages, you can use this feature to create a summary page that lists (and links to) of all pages based on a selected document type.

For example, you could create a "News" document type. You then allocate it a default storage directory, in this case "news", and a template called news_template.tmpl. Now, whenever you wish to create a new news page, you only need to select the "News" document type – both the template and the default storage directory for news pages will be selected automatically.

3.6.1.1 Creating a new document type

In the *Options* drop-down menu, select *Document Types...* The *Document types* dialog box appears (see Figure 103).

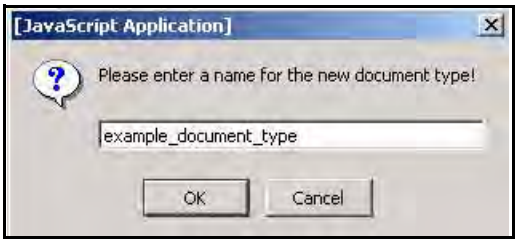
Figure 103 Creating a new document type



To create a new document type, click the *New document type* button.

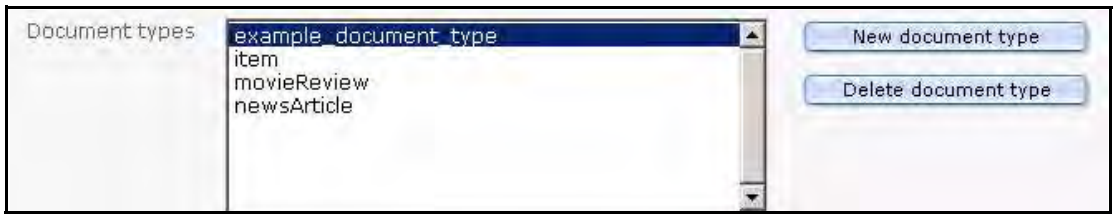
A response box opens where you are asked for the name of the new document type. Enter the name and select *OK* (see Figure 104).

Figure 104 Naming a new document type



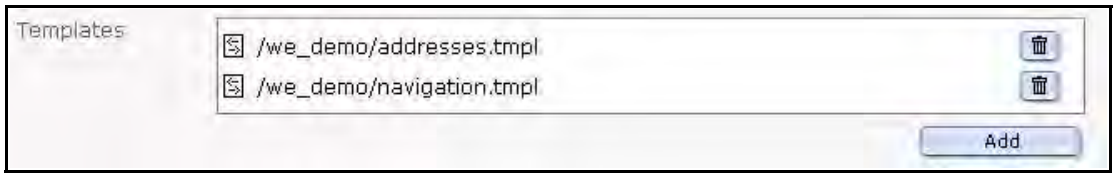
The new document type will appear in the list box as shown in Figure 105.

Figure 105 New document type/delete document type



The *Templates* area allows you to allocate the appropriate template(s) to the active document type, displayed in the *Name* area. Click on the *Add* button to browse your files for a template (see Figure 106).

Figure 106 Selecting a template for a document



In the *Default values* area, specify the default settings for the document type.

- *Directory*. Use the *Select* button to choose a directory for all webEdition pages based on this document type.
- *Subdirectory*. Use the pick list to create a subdirectory which is identified in your explorer menu by the date of creation. Thereafter all documents in this document type will be associated with the default creation date.
- *Default extension*. Use the pick list to select the default extension for all documents in this document type.
- *Template*. Use the pick list to select or de-select a default template.
- *Generate dynamic page*. Use this checkbox to choose whether or not you want the page to be dynamically created.
- *Is searchable?* Use this checkbox if you wish your page to appear on summary pages.

  Note: If a page is not searchable, it will not appear on summary pages.

- *Category*.

Once you have specified all the settings, click *Save*.

3.6.1.2 Deleting a document type

You can delete a document type by clicking an item in the list box located in the *Document type* dialog box and then clicking the *Delete document type* button (see Figure 105, "New document type/delete document type" on page 91).

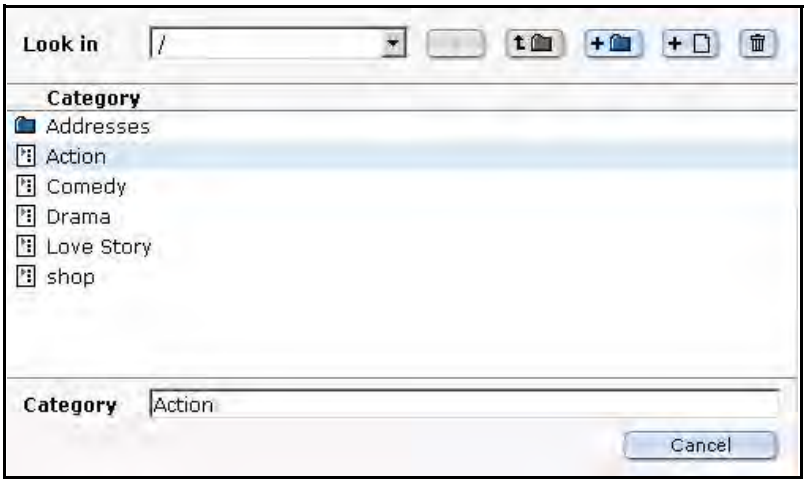
3.6.1.3 Editing Document types

In the *Document types* section, select the document type you want to edit and follow the procedure described in Section 3.6.1.1, "Creating a new document type" on page 90.

3.6.2 Categories

The *Categories* feature is accessed by selecting *Options* > *Categories* in the main menu (see Figure 107). The *Categories* feature offers an alternate method of generating summaries. You can also use *Categories* together with *Document types* to generate sub-summaries.

Figure 107 The *Categories* feature



For example, if you have a "News" document type, you could subdivide it into “Sports”, “Politics”, “Computing” and “General” so that you can generate summary pages such as “Sports News”. Thereafter, you could create sub-categories that are subordinate to the Sports category such as “Football”, “Baseball”, “Hockey”, etc.

3.6.2.1 Creating a new category

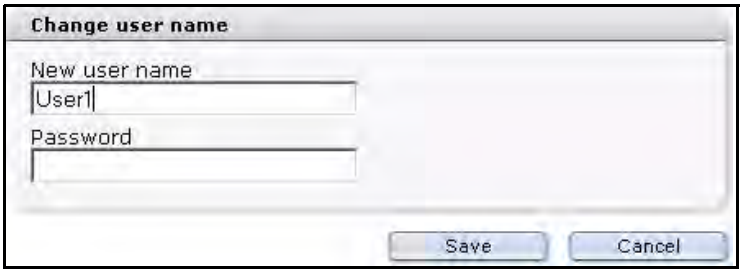
Select *Options* > *Categories*. A file management window opens that allows you to create or delete categories and folders containing categories. This dialog box and its icons operate in the same manner as the *Browse Server* dialogue box (see Section 2.3.1, "The File menu" on page 38, for further details on how to use webEdition’s file management windows).

3.6.3 Changing user name and/or password

You can change your user name by clicking on *Options* > *Change user name.* The *Change user name* screen appears (see Figure 108):

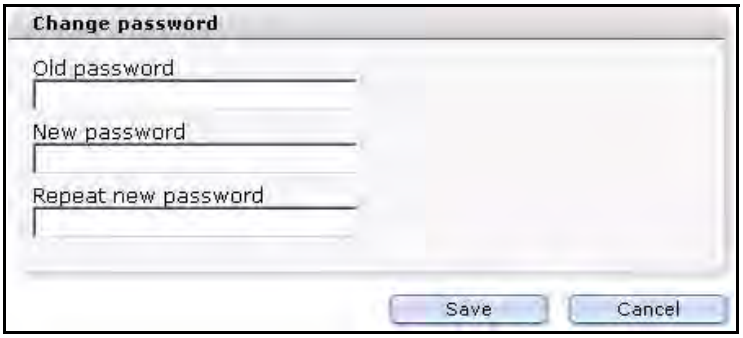
> ***Note:*** If you have installed the User Management Module, you will see *User Management* in the menu. Click on this item to access the module and to perform user management functions. For details see *The User Management Module User Guide*.

Figure 108 Change user name screen



To modify your password, select *Options* > *Change password*. Enter your old and new passwords in the designated fields, then click on the *Save* button (see Figure 109).

Figure 109 Change password screen



3.7 How an editor works within a template: using template input fields, lists and menus

Whether you are creating a new document or modifying an existing one, you will be working within a structure that is defined by a template. When you open or create a document in the *Edit* view, you see various fields, pick lists, or menus which you can fill with text, images, etc. This editing mode is your working interface; what you see in this view is not what the finished page will look like. To see how your page will look to visitors to your Web site, select the *Preview* tab. The following table summarizes the fields that commonly appear in the templates you will be using. The sub sections thereafter describe the fields in detail.

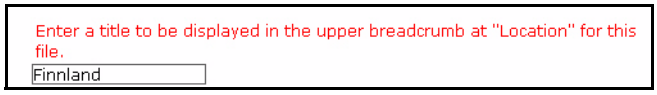
Field	Description	we:tag
Text input	A single input box.	we:input (text)
Date selection	A drop-down menu to enter date and time.	we:input (date)

Field	Description	we:tag
Check box	A check box. Toggles a field on or off	we:input (checkbox)
Input choice box	A normal single-line input box with a drop-down menu beside it.	we:input (choice)
Text block	A multi-line input box.	we:textarea
Select box	A select box (also called a pick-list).	we:select
List block	Used to repeat blocks of fields	we:block
List of links	An expandable list of all your links.	we:linklist
Images	Allows you to insert images	we:img
Movies	Allows you to insert Flash or Quicktime movies	we:flashmove we:quicktime

3.7.1 Single-line input box (<we:input type="text">)

This field is used for single-line text blocks such as headers. You can also write HTML code in the input box (see Figure 110).

Figure 110 Sample page of single-line input box in the *Edit* view

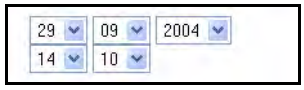


Sample page: /we_demo/news/2004/finnland.html

3.7.2 Date (<we:input type="date">)

By clicking on these fields, you can set the date you want to appear on the Web page (see Figure 111).

Figure 111 Sample page showing date fields



Sample page: /we_demo/news/2004/finnland.html

3.7.3 Checkbox (<we:input type="checkbox">)

This field is used as a switch to insert or remove an area of a template.

3.7.4 Input choice box (<we:input type="choice">)

This feature allows you to create a single-line input box beside a drop-down menu (see Figure 112).

Figure 112 Using an input choice box

When you select an entry in the drop-down menu, the system copies your choice to the input box (on the left). You also have the option of typing your own value in the input box, if you require a value different from those in the pre-defined drop-down menu. The values available in the drop-down menu are pre-defined by your template administrator. For example, the template code seen in Figure 112 is as follows:

```
<we:input type="choice" name="City" size="40"
values="Berlin|Madrid|Ottawa|Washington"/>
```

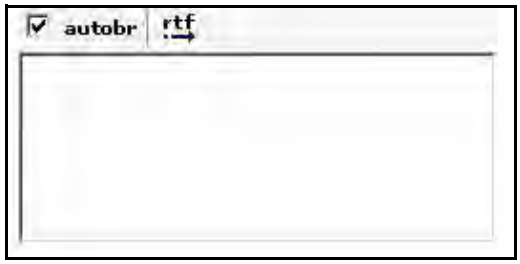
3.7.5 Text box (<we:textarea>)

The appearance of this text box is determined by the browser version you are using. We recommend using Internet Explorer 5.5 or higher, since it displays the text area as a WYSIWYG editing program if the *dhtmledit* attribute in the template is set to *ON*.

3.7.5.1 Normal input screen

This text box is intended to be used for relatively long texts. The text can be formatted or links can be inserted using HTML commands (see Figure 113).

Figure 113 Using a text box



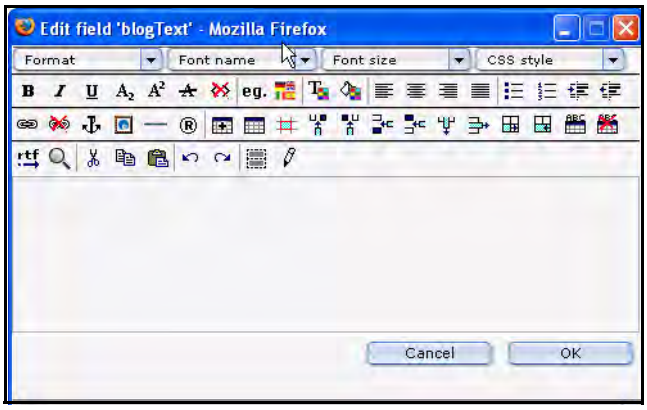
If the *autobr* (automatic line break) attribute has been specified (true or false), a checkbox will be displayed at the top of the input box. You can turn the *autobr* feature on or off using this checkbox. If *autobr* has been activated, all forced line breaks (return key) will be translated as HTML line breaks (<br>). If *autobr* has been set to *false*, the return key is equivalent to a space. For information on the .rtf function, see Section 3.7.5.4, "Importing Rich Text Format files (*.rtf)" on page 100. The code for Figure 113 is as follows:

```
<we:textarea name="Content" width="250" height="100" autobr="true"
wysiwyg="false" importrtf="true"/>
```

3.7.5.2 WYSIWYG input screen

If your template administrator has included a WYSIWYG editing screen in your template (by setting the we:tag attribute *wysiwyg* to *true*), the text box appears in the form of a WYSIWYG (What you see is what you get) text editor (see Figure 114).

Figure 114 Using a WYSIWYG input screen



You can edit texts in this field just as you would in a standard word processing program.

If you wish to work directly with HTML tags in the input box, you can do so by clicking the toggle that appears as a pencil icon directly above the input box. If you do so, the pencil icon will toggle to an eye icon, and the input box will allow you to enter HTML code (see Figure 115).

Figure 115 WYSIWYG toggle buttons

3.7.5.3 Accessibility functions of the WYSIWYG Editor

webEdition supports functions in its WYSYWIG editor that make it easier for you to create Web pages that are accessible to the visually impaired. (see Figure 116).

Figure 116 WYSIWYG textfield

3.7.5.3.1 Features for screen readers

- Expanded abbreviations for screen readers:

  In order for a screen reader to read an abbreviation aloud, the abbreviation must be expanded. Highlight the abbreviation that appears in the text area, and click the "eg." button from the WYSIWYG editor. The following window opens:

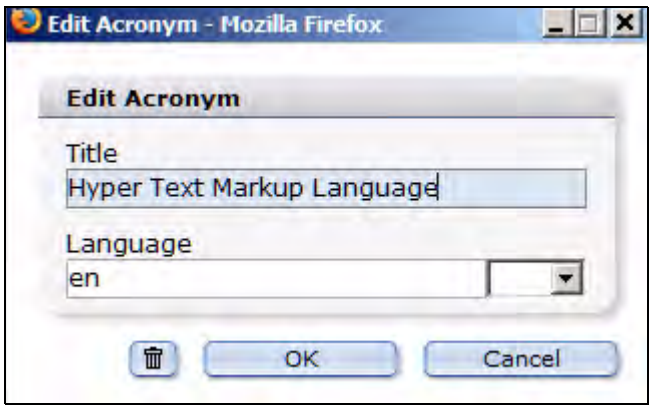


Figure 117 Define abbreviation for a screenreader

After highlighting the abbreviated term, the expression is entered in full in the *title* field. The language of the expression is defined by a 2-character country code in the *language* field. In our example, the definition for the abbreviation "HTML" is shown.

- Language selection for screen reader:

  To enable screen readers to read foreign words correctly (that is, those that are not in the default language of the document), highlight the respective word and click on the *language* button (which appears as a group of flags). The following dialogue opens:

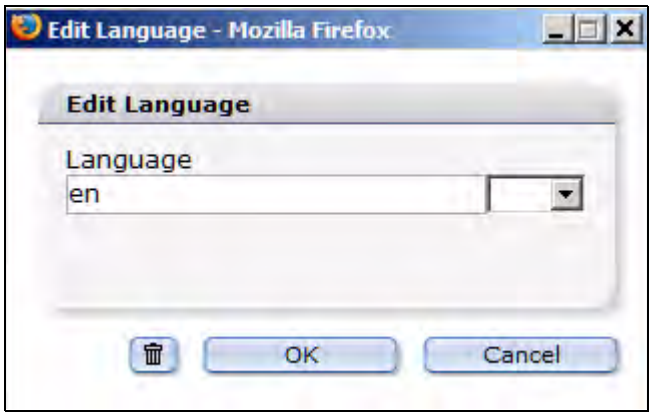


Figure 118 Language selection for screenreader

From the dropdown menu, select the two character international countrty code representing the language of the foreign term (for example, "de" for German, or "en" for English).

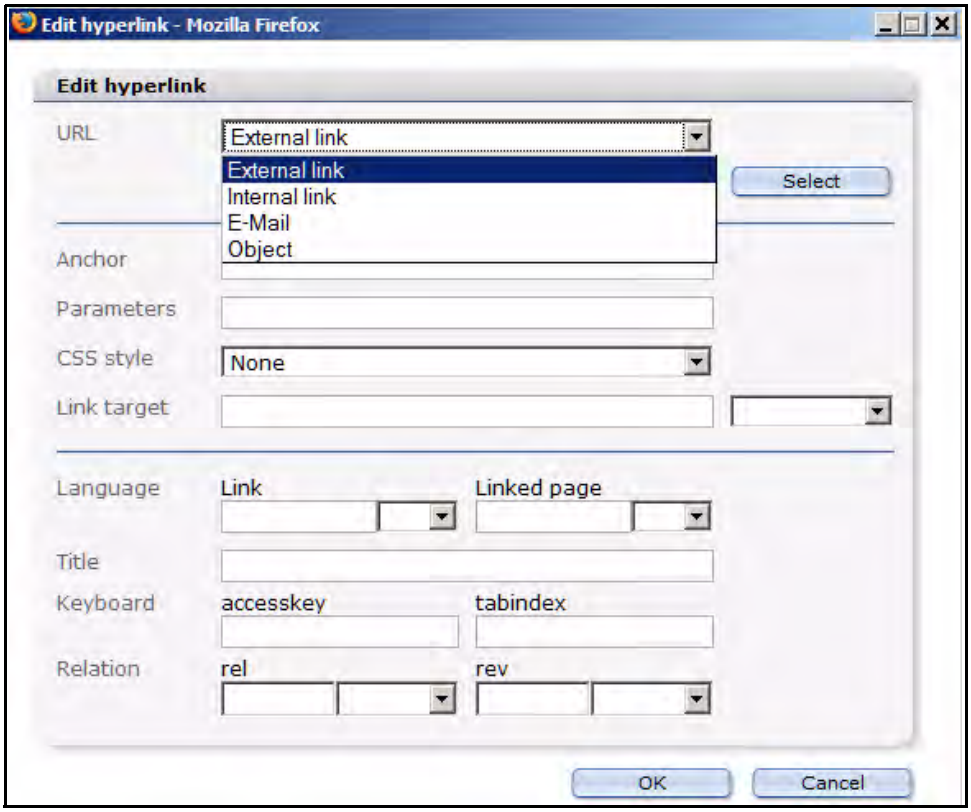
- *Edit Hyperlink* dialogue enhancements for screen readers

  Additional fields appear in the Edit Hyperlink dialogue box beginning in Version 3.2 (see Figure 119 on page 98). To activate the hyperlink dialogue box, highlight the text that you wish to link, and click on the insert/edit link button (the chain icon). You can now set the language of the linked text as well as the language of the linked page (*Link* and *linked page* fields under *language*). These fields correspond to the HTML attributes *lang* and *hreflang*.

  The fields *title*, *accesskey* and *tabindex* have been added. Under *accesskey* a character or a figure can be entered with which a link can be activated. The link can also be activated with ALT+ character/figure.

For *Tabindex*, a figure can be entered defining the order in which the links are navigated by pressing the *tab* key. Enter the title of the link in the *title* field. When the mouse is placed over the link, this title will be displayed in text (as a mouse-over) in most browsers.

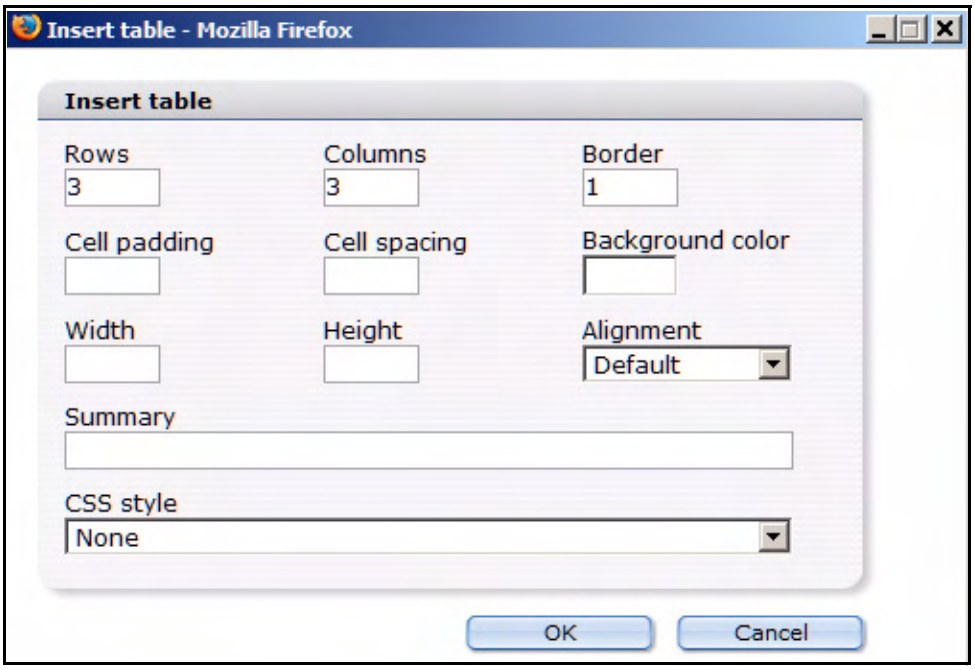
Figure 119 Edit Hyperlink in WYSIWYG



- Table *Summary* field for screen readers

  If you click on the *Insert table* button in the WYSIGYG editor, you can add a summary for a table in the *Summary* field of the *Insert table* dialogue box (see Figure 120). A screen reader will read the summary as a description of the table.

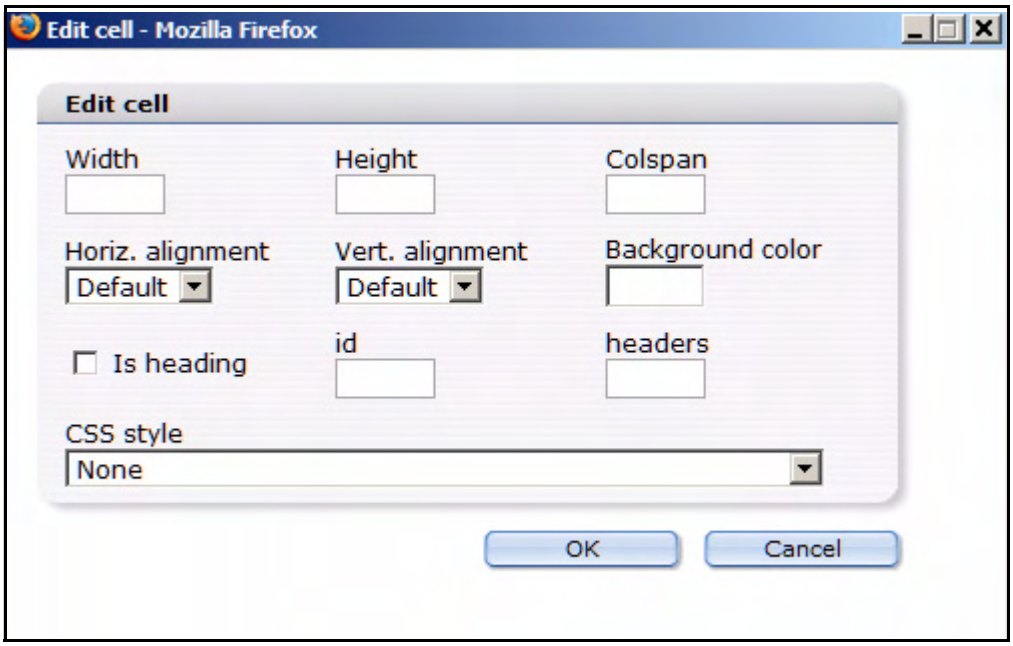
Figure 120 Insert table - table summary



- Editing table cells for screen readers

  By clicking the *Edit table cell* button, table cells can be defined as headline cells (*th* instead of *td*) in the *Edit cell* dialogue box (see Figure 121). To identify a cell as the header, activate the *is headline* checkbox. Additionally, the attributes *id* and *headers* have been added for non-visual output (such as screenreaders), so you can explicitly reference a data cell to one or multiple headline cells. The sytem is thus able to repeat the content of the headline cell(s) for every data cell. Otherwise, it would be very difficult to establish a relationship between a data cell and a headline cell for the voice output, as voice output lacks the the visual cues inherent in a table. To set these references, you must set a unique name for the *id* of the headline cell. In the data cells, you must enter this *id* in the *headers* field. Multiple *ids* are separated with spaces.

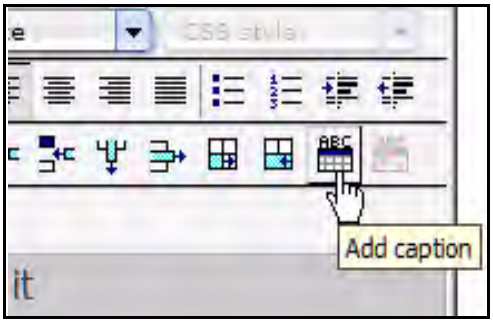
Figure 121 Edit table cell dialogue box



- Table captions for screen readers

  After creating a table, a table caption can be added (or removed) by clicking the *caption* buttons. The caption will appear in the WYSIWYG editor field above the table.

Figure 122 Table caption



3.7.5.4 Importing Rich Text Format files (*.rtf)

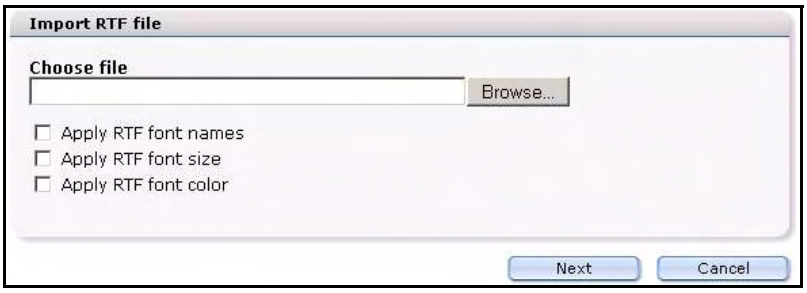
You can import RTF files directly to either the normal or the WYSIWYG text input boxes. This means that you can write and format your text in a standard word processing application. Therafter, you can import your text using the RTF import function and webEdition will retain your original formatting.

Note: The import button appears only if the *importrtf* attribute has been set to *true* in the template by the template administrator.

Figure 123 Import .rtf file button

By clicking on the *Import .rtf file* button (Figure 123), you open the following dialogue box (see Figure 124):

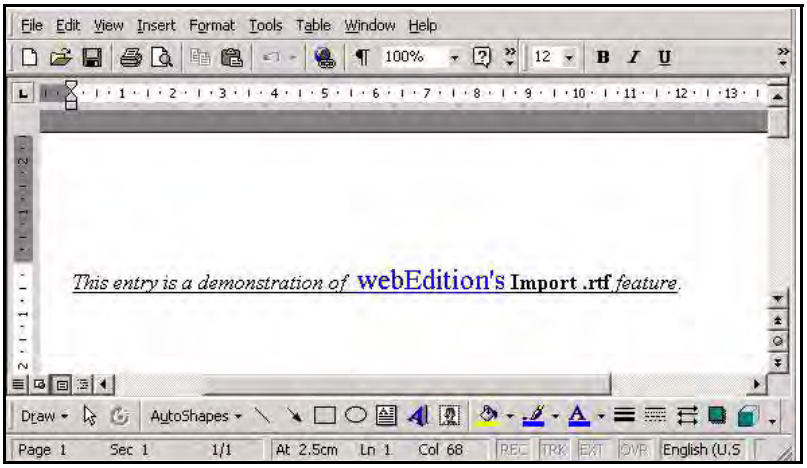
Figure 124 The Import RTF file dialogue box



Locate the file you wish to import from your local hard drive by clicking the *Browse* button and then upload it to the preview field by clicking the *Import* button. All the formatting will be applied. Since there are many different RTF formats, only the font, font size and font color are not automated and thus optional.

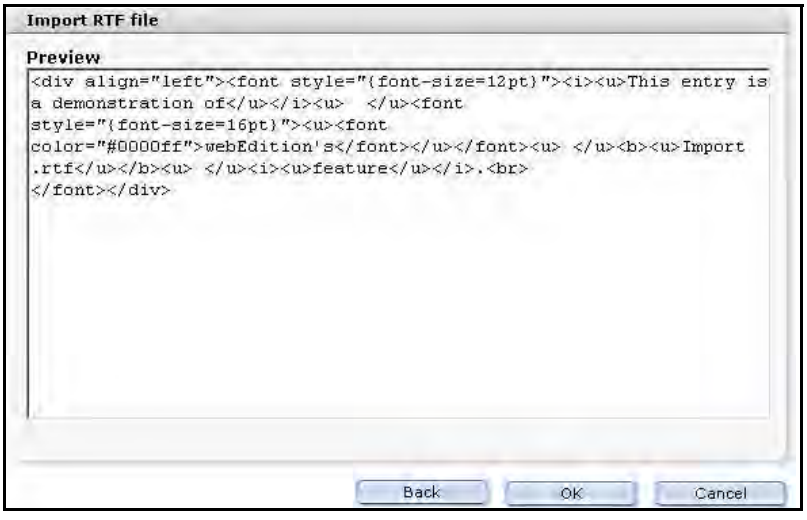
Figure 125 shows an example of text to be imported from a word processing program.

Figure 125 Example of an RTF file



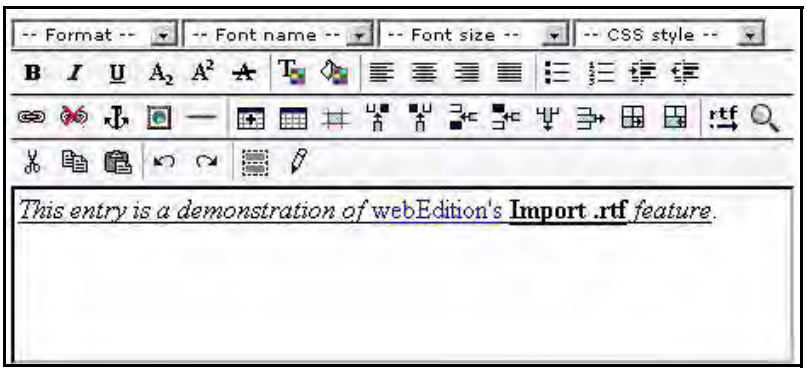
Thereafter, the following content is seen in the *Preview* area (see Figure 126):

Figure 126 RTF text in *Preview* mode



Click *OK* to inserted the imported text into the text box.

Figure 127 RTF text imported into webEdition

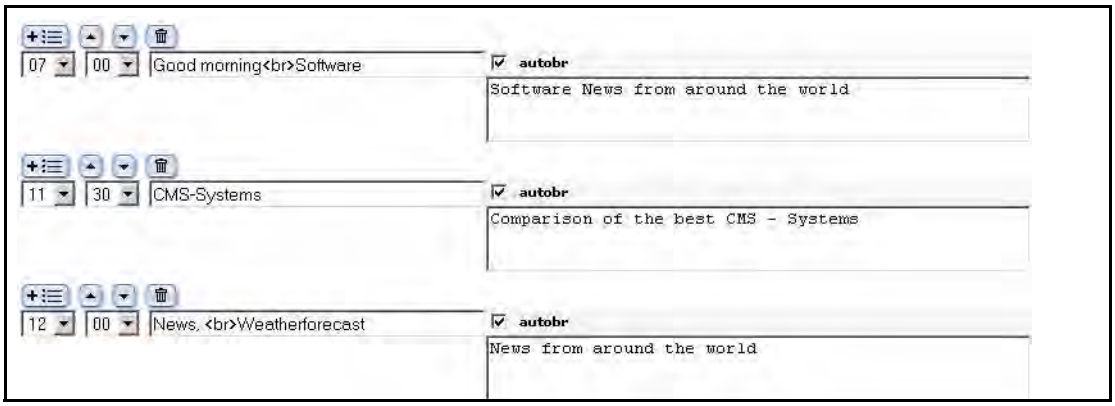


Note: RTF formats may differ from one application to another. Such differences may distort the way the text is displayed in webEdition. Such errors can be corrected in the *Edit* view.

3.7.6 Lists (<we:block>)

Lists are blocks of code which can be modified using the editing controls. You can duplicate a list item by clicking on the plus button. By clicking on the arrow icons, you can move a block up or down within the list. You can delete a block by clicking on the adjacent trash can icon. An example of a list block is shown in Figure 128.

Figure 128 Example of a list block



When viewed in the preview view, the list block looks thus:

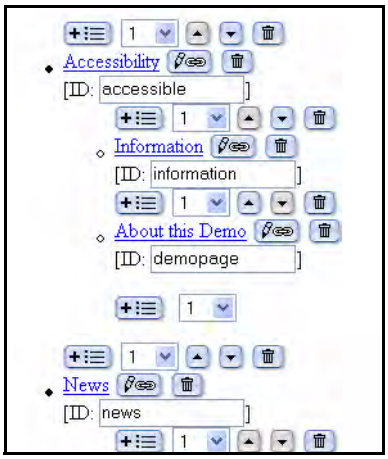
Figure 129 Example of a list block in *Preview* view

07:00	Good morning Software	Software News from around the world
11:30	CMS-Systems	Comparison of the best CMS - Systems
12:00	News, Weatherforecast	News from around the world

3.7.7 Link list (<we:linklist>)

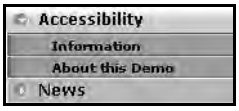
You can use the link list feature to create link lists (see Figure 130). A plus button will appear, which you can click to add a new link to the list. Links can be deleted by clicking the trash can icon.

Figure 130 Example of a link list (*we_demo/includes/navigation.php*)



When viewed in the preview view, the list block looks thus:

Figure 131 Example of a link list in *Preview* view



The following procedure describes how to create and edit hypertext links. For the purpose of this procedure, see we_demo/links/links.html on the webEdition demo page.

Procedure 7 Adding and Editing links

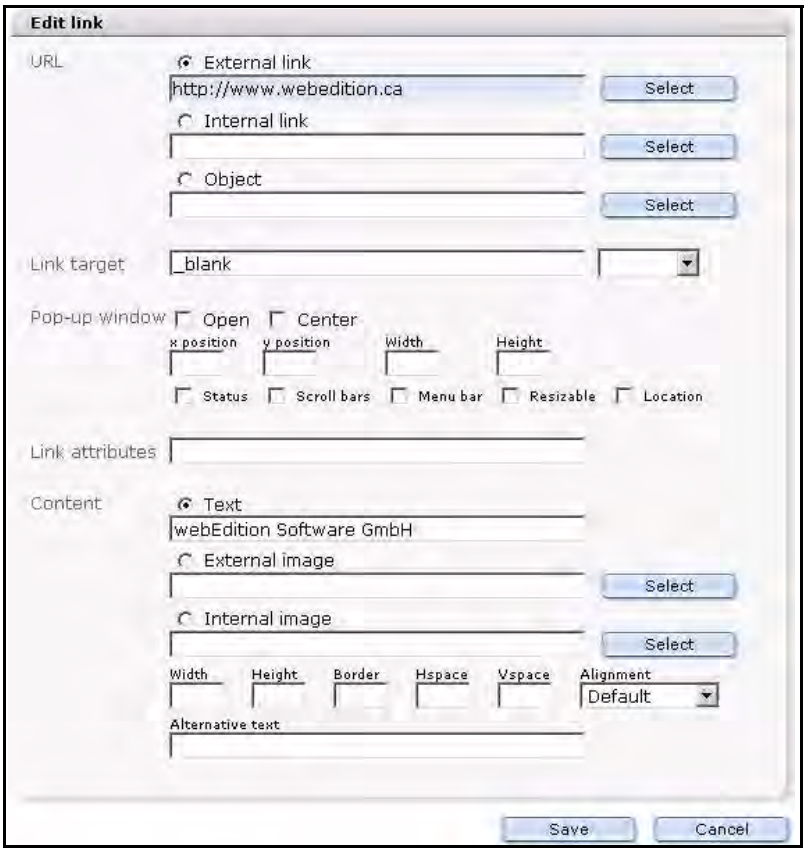
In the Documents Edit view

1 Open the file we_demo/links/links.html, and click on the *edit hyperlink* button (see Figure 132).

Figure 132 Edit hyperlink button

The Edit link dialog box opens (see Figure 133).

Figure 133 Edit link dialogue box



2 In the *URL* area, specify whether the link is to be to an internal or external page or object by ckicking one of the radio buttons.

3 Click the *corresponding Select* button to launch the file manager, where you can browse for and select your target document.

4 If you use frames, you can specify the target frame (_name-targetframe) in the *Link target* field. The select box to the right contains some standard target designations.

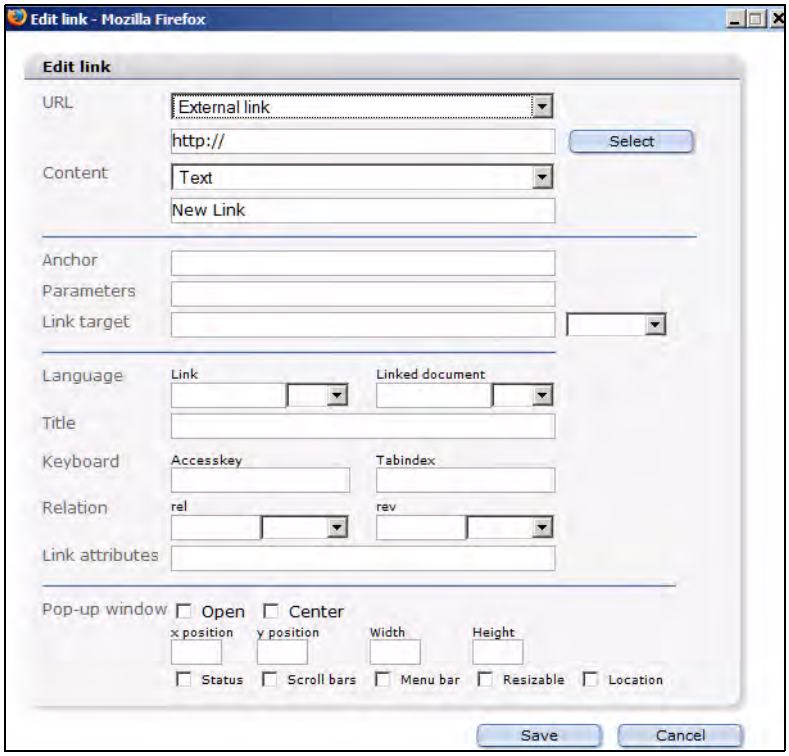
5 In the *Content* field, enter a text string that will appear as the link on your Web page. If you would rather link to an image, you can specify either an internal or external image as well as where it is to be positioned and the alternative text. You can also select an internal image using the file manager, which can be launched in the *Content* area by clicking the *select* button.

6 You have completed this procedure.

3.7.7.1 Accessibility enhancements for links

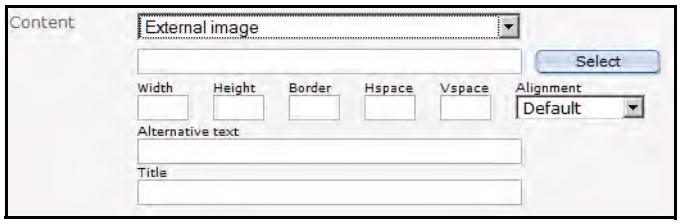
The link dialogue has been expanded (see Figure 134). The new fields improve your ability to create accessible Web sites. For more information about thes fields, see Section 3.7.5.3, "Accessibility functions of the WYSIWYG Editor" on page 96.

Figure 134 Edit Link



When linking images, you can also enter *Alternative Text* as well as a *Title* that will be read by screen readers.

Figure 135 Linking images in version 3.2



3.7.8 Select box (<we:select>)

A select box (or pick-list) allows you to choose one item from a list of pre-defined values (see Figure 136).

Figure 136 Select box

When working in the *Edit* view, click the arrow beside the box to view the values that are available to you. The value you choose will be displayed on the finished Web page. The template code is thus:

```
<we:select name="Cities" size="1">
  <option>London</option>
```

```
    <option>Paris</option>
    <option>New York</option>
    <option>Toronto</option>
</we:select>
```

3.7.9 Image field (<we:img>)

An image field allows you to insert an image in the page content (see Figure 137).

Figure 137 The image field

By clicking on the edit button, you will launch a file manager in which you can select an image that has been uploaded to webEdition. For further details, see Section 3.4, "Working with graphical documents" on page 73.

The code for Figure 137 is: `<we:img name="Picture"/>`

3.7.10 Movie fields (<we:flashmovie> and <we:quicktime>)

The movie fields allow you to insert a Flash or Quicktime movies in the page content (see Figure 138).

Figure 138 The Flash and Quicktime movie fields

By clicking on the edit button, you will launch a file manager in which you can select a Flash movie that has been created or uploaded to webEdition. The template code for Figure 138 is:

```
<we:flashmovie name="MyFlash" width="100" height="100"
user="editor1,editor2"/>
```

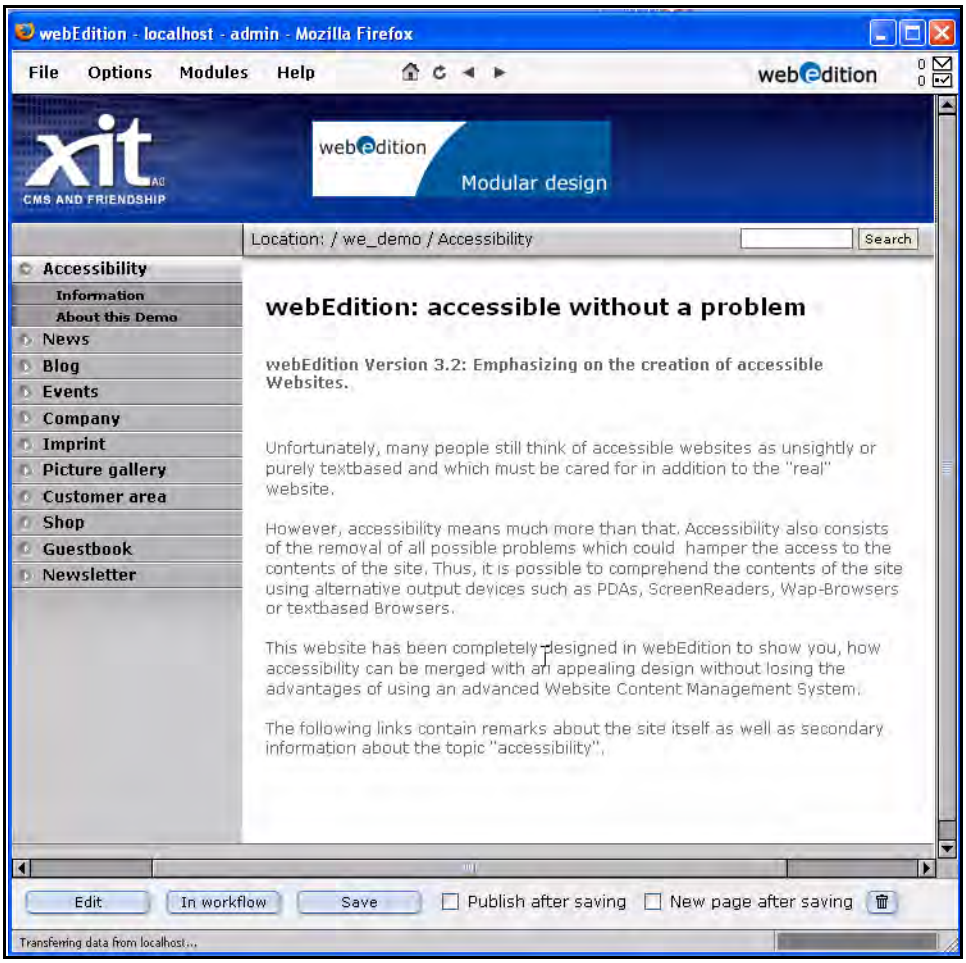
For more information on Flash movies, see Section 3.5.2, "Using Flash movies" on page 86; for Quicktime movies, see Section 3.5.3, "Using Quicktime movies" on page 87.

3.8 Editing documents in seeMode

seeMode is designed to make the editing of Web pages simple and efficient. This mode offers you a WYSIWYG view of your Web site. You can navigate through your site just as you would with your regular browser (see Figure 139). When you get to the page you wish to edit, you simply click the *Edit* button at the bottom of the page.

Note: The seeMode feature in webEdition 3.0 cannot be used with Web sites that use HTML frames.

Figure 139 The seeMode preview



Those fields on your page which represent included files that can be navigated to and then modified are identified by a seeMode edit icon.

Figure 140 The seeMode included file list

The following procedure describes how you can use the seeMode to edit a document.

Procedure 8 Editing a document using seeMode

At the main webEdition login screen

1 Select the seeMode radio button, enter your user name and password (see Figure 141).

Figure 141 Login to seeMode

2 Click *Login*.

webEdition opens in seeMode at your designated home page (see, for example, Figure 139, "The seeMode preview" on page 107). The start page is determined in the Preferences screen (see Section 2.2.1, "The User interface view" on page 29). Your ability to change this start page is dependent on the permissions that you have been given by your administrator.

3 You have two choices. You can select *Edit* or *In Workflow*.

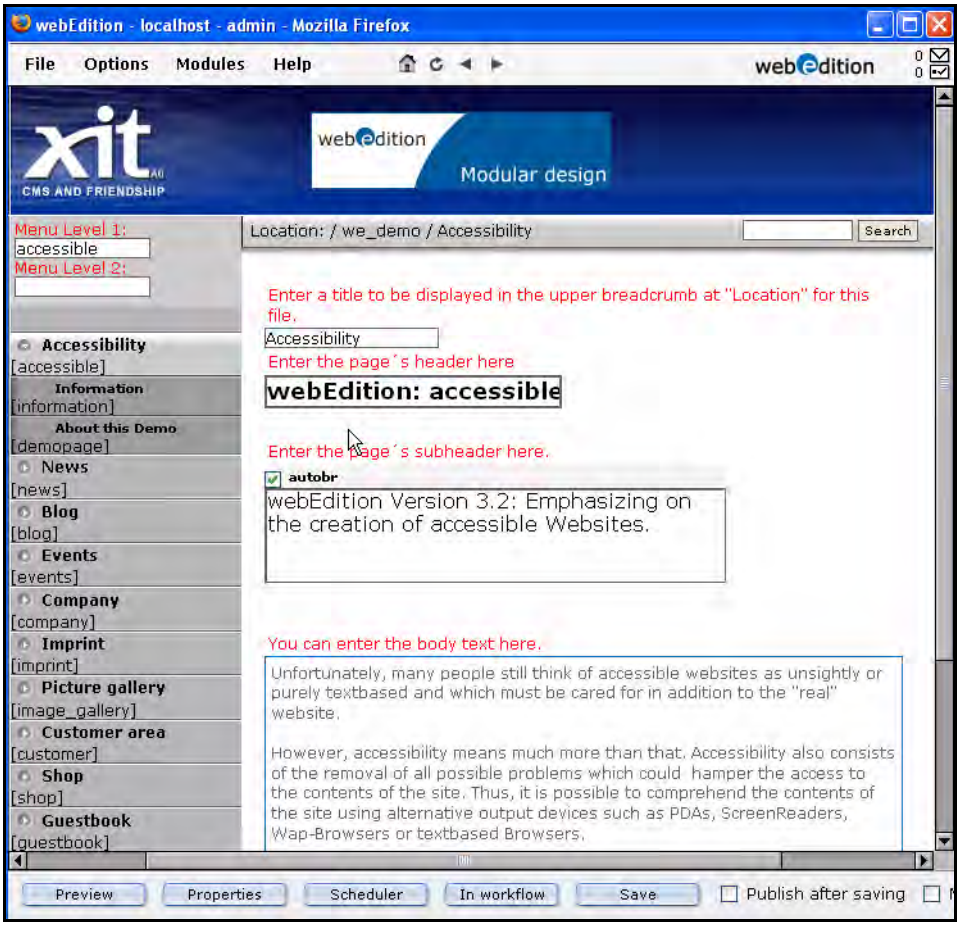
a If you have installed the Workflow Module, you have the option of placing your document into the workflow.

Consult the documentation for the Workflow Module, then go to Step 6.

b If you wish to edit the page, click the *Edit* button.

The page opens in the seeMode edit view and you can modify the fields (such as those described in Section 3.7, "How an editor works within a template: using template input fields, lists and menus" on page 93; see also Figure 142).

Figure 142 seeMode edit view



4 Edit your content by changing the data in the available fields.

5 Choose one of the following options:

- **a** Return to the preview screen (by clicking the *Preview* button).
- **b** Edit the properties of the page by clicking the *Properties* button (See Section 3.1.1 for further details).
- **c** Place your document into the workflow.
- **d** Save your changes (with or without selecting the *Publish after saving* option).

  After saving your file, webEdition will confirm the action, the edit window closes, and you are returned to your designated home page.

6 You have completed this procedure.

3.9 The navigation tool

With the navigation tool you can simply and comfortably control the navigation of your web page.

3.10 Using the navigation tool

Launch the navigation tool with *Options* > *Navigation...*.

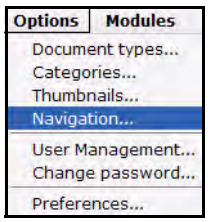
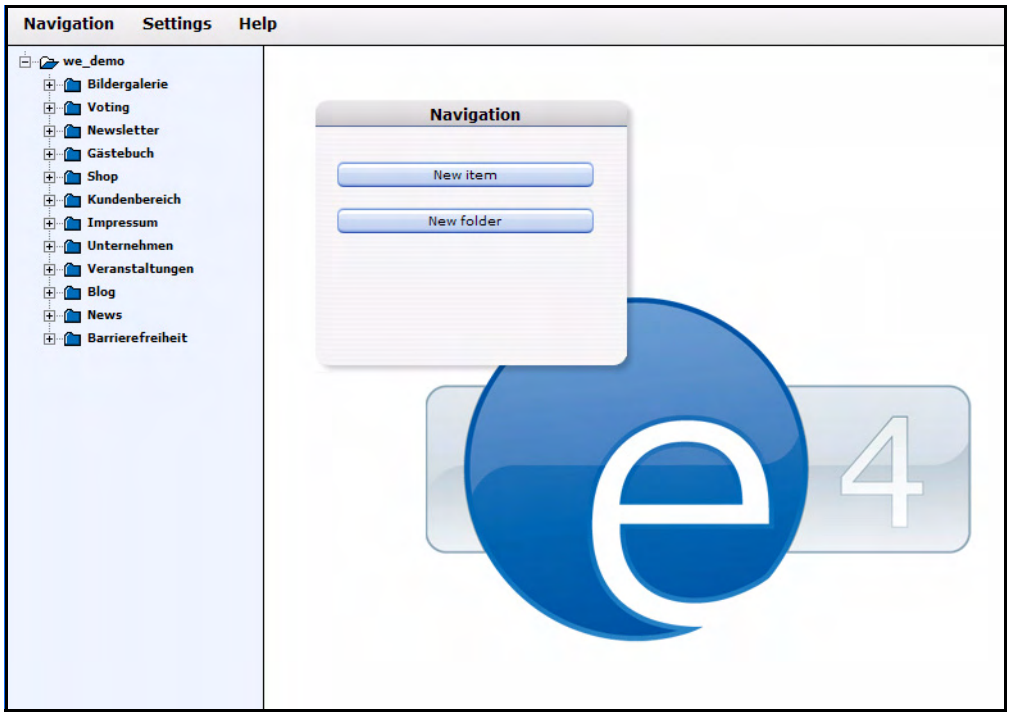


Figure 143 Menu navigation tool

As a result the navigation tool is opened in a new window (see Figure 144). On the left side of the window, the navigation is illustrated in a folder structure. Concentric you find the Quickstart entries *New entry* and *New folder*.

Figure 144 The naviagtion tool



In the upper margin there are the menu items *Navigation*, *Preferences* and *Help*.

3.10.1 The menus

The *Navigation* menu contains the following items:

- *New* > *Entry/Folder*: Select this item to create a new menu entry or folder. This corresponds to the functions of the Quickstart buttons
- *Save*: Saves the currently selected navigation entry or folder
- *Delete*: Deletes the currently selected navigation entry or folder
- *Quit*: Closes the navigation tool

The menu *Settings* contains the menu item *Rules for highlighting*. This will be explained in Section 3.11.

The menu *Help* contains the following items:

- *Help...*: Leads to the webEdition online help
- *Info*: Displays the license information and version number

3.10.2 Creating new navigation items

In the following, the use of the navigation tool is explained.

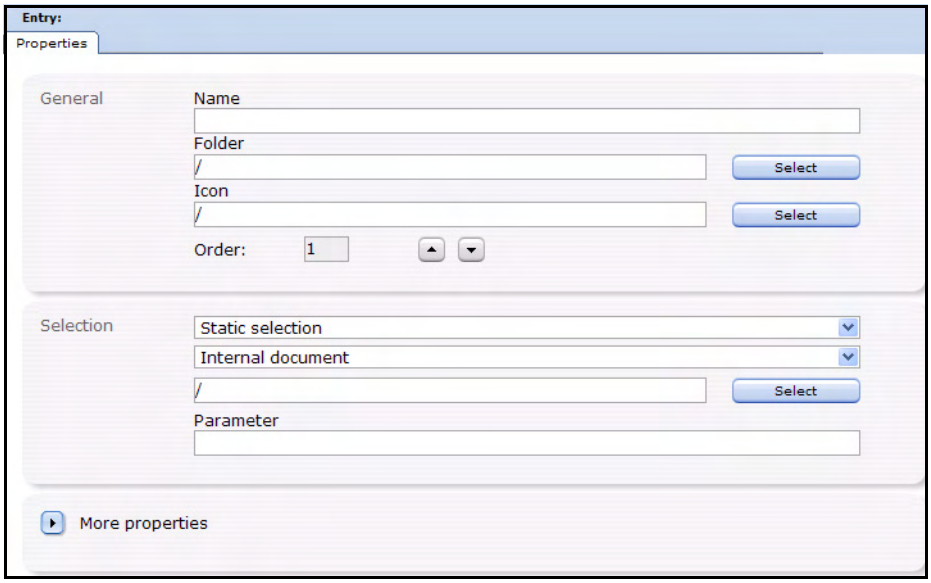
> ***Note:*** The examples and screenshots in this chapter correspond to the we_demo which installs with webEdition.

On the left side of the navigation tool you can see the navigation (in this case we_demo) which finds its correspondance on the website.

Figure 145 Navigation in the navigation tool and on the website

Choose the quickstart button *New item* or *Navigation* > *New* > *Entry* and the *Properties* view for the new entry is openend.

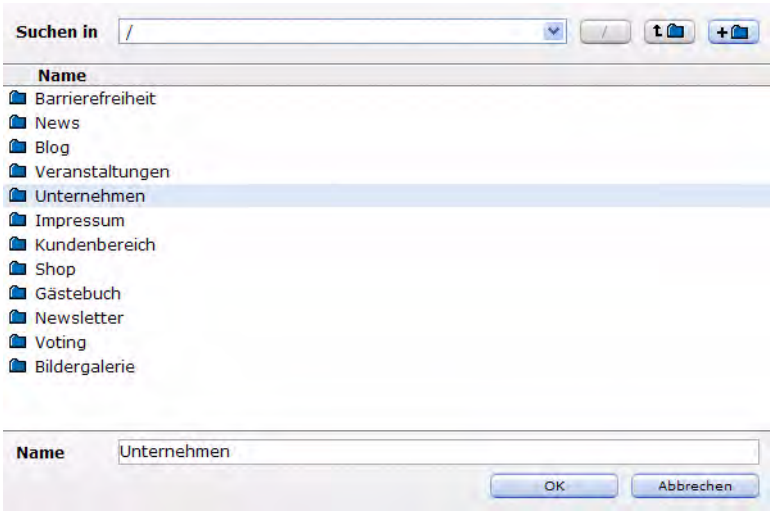
Figure 146 View *Properties* for new entry



In the area *General*, the following fields are located:

- *Name*: Set the name of the entry to the name to be displayed in the navigation
- *Folder*: Click the *Select* button to open a window where you can choose from the existing navigation folders. The new entry will then be created as a subnavigation item.

Figure 147 Select navigation folder



- *Icon*: Here you may select an icon for your navigation entry. The icon must have been imported before in webEdition.
- *Order*: Here you can change the order of the naviagtion entries.

The content of the area *Selection* differs according to selected choice (Static /Dynamic selection; type of entry)

3.10.2.1 Static navigation entry

The following navigation entries can be linked for a static entry:

Figure 148 Selection navigation entry

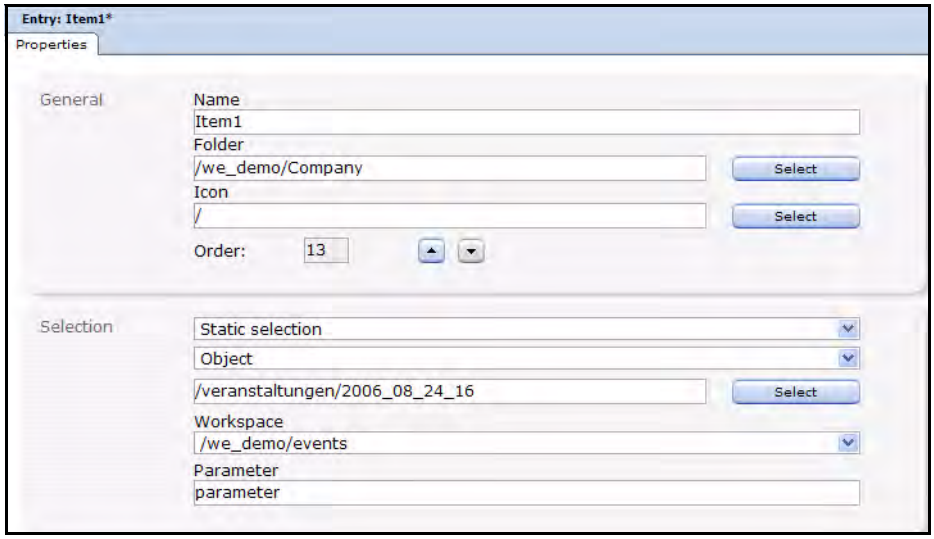
- *Internal document*: Use the *Select* button to link any webEdition document with the navigation entry.
- *External documents*: Enter a valid web link after the http:// to which the navigation entry should link.
- *Object*: Use the *Select* button to link any webEdition object with the navigation entry.

  Note: Please note that workspaces have to be defined for objects! In the generated link the ObjektID will be passed; the object itself can be displayed in any dynamic webEdition document.

- *Category*: Select a category to be displayed in the navigation. With *Name of the category parameter* a user defined parameter can be passed with the link. As a concrete application example of such passed category parameters the webEdition shop demo can serve, e.g., the navigation under the URL *http://demo.webedition.de/shop_demo/*

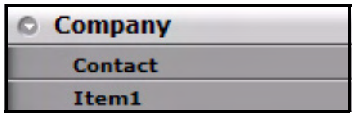
In all these cases you can define in the field Parameter, in addition, a parameter to be passed which should be transmitted in the link of the navigation.

Figure 149 Example entry



The example entry in Figure 149 would result in the link *http://myPage/we_demo/events/index.php?we_objectID=13&parameter* and would be a subnavigation item of the menu item *company*.

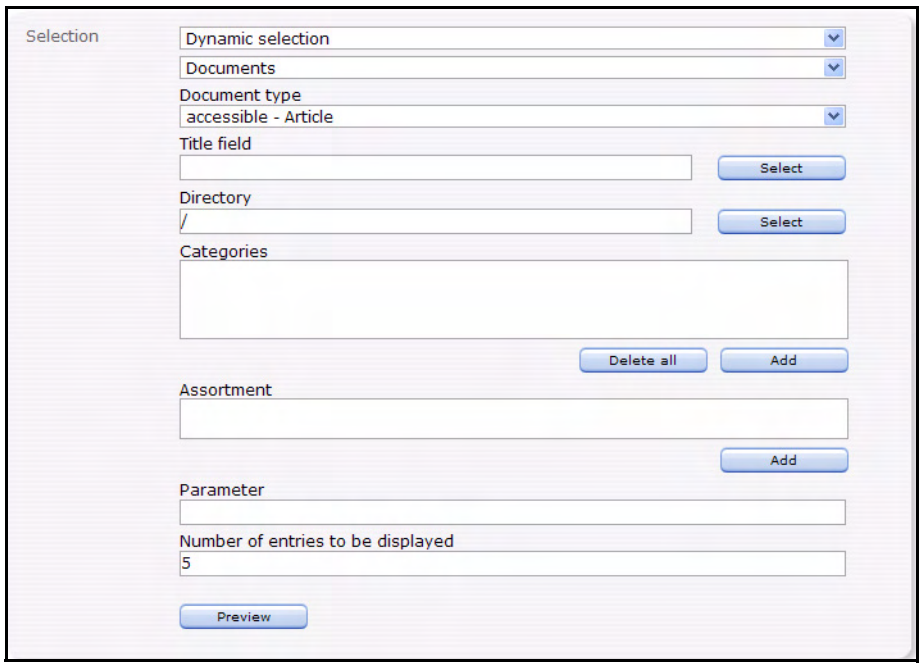
Figure 150 Subnaviagtion item



3.10.2.2 Dynamic navigation entry

Select in the upper Dropdown menu of the *Selection* area the *Dynamic selection* to bring up advanced possibilities for an automated updating of the navigation. Select at first the kind of the dynamic entry: *Documents*, *Objects* or *Categories*.

Figure 151 Area *Selection* - Dynamic selection



- *Documents*: Here you determine by the choice of the document type which documents should be linked in the navigation. The naming of the links in the navigation is applied by the *Title Field*. The possible selections in the *Title Field* depend on the template defined in the document type! With *Directory* and *category*, you may limit the documents to be displayed.

  In the field *Assortment*, you can determine the order of navigation entries by a certain field (here, too, the fields are taken from the document type). The *parameter* Field can be filled with a parameter to be passed. The number of *entries to be indicated* limits the amount of the indicated navigation items.

  Click the *Preview* Button to open a preview window in which the names of the generated entries as well as their internal links are indicated. In the example in

Figure 152 you see the preview for the navigation entries of the document type “LatestNews“ with the title field “HeadlineKurz“

Figure 152 Preview of the navigation entries

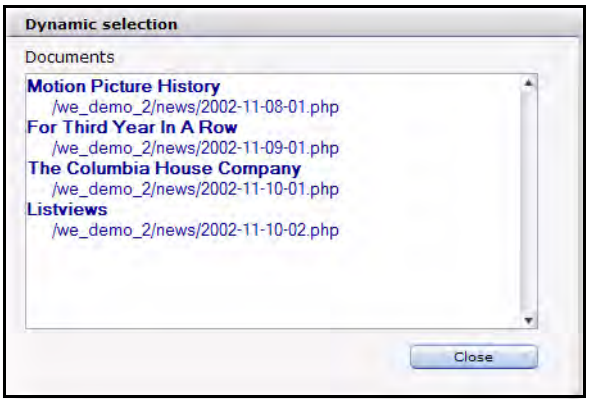


Figure 153 Navigation on the website

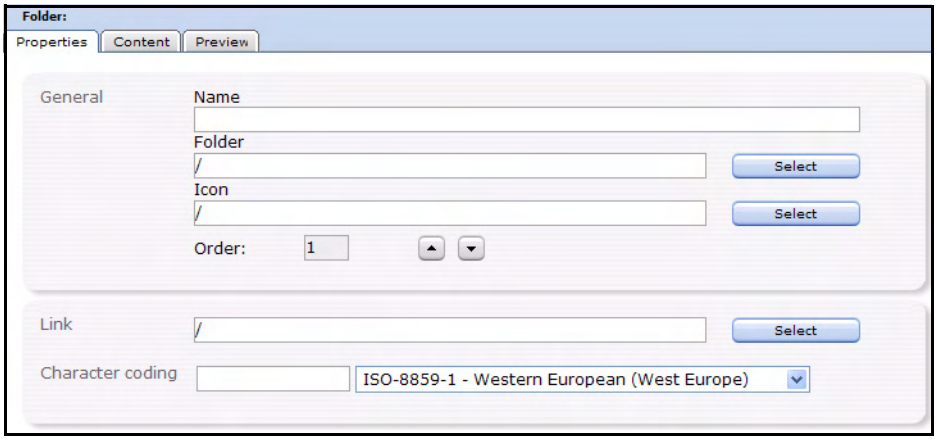
In the navigation on your website the new dynamic entry is diplayed like shown in Figure 153. In this case, the number of entries to be displayed was limited to “3“.

- *Objects*: Select in the *Class* dropdown menu an existing class. Please note that a valid workspace has to exist! The remaining fields are filled analogously to the dynamic documents. The selections *Title field* and *Assortment* hereby result from the fields defined in the class.
- *Categories*: The dynamic navigation entries for categories are used analogously to those for static categories.

3.10.3 Creating a new navigation folder

To create a new navigation folder, click *Navigation* > *New* > *Folder* in the menu of the navigation tool window or the Quickstart button *New Folder*.

Figure 154 New folder



Enter the following data in the *Preferences* view in the *General* area:

- *Name*: Enter here a name for the folder which will also be displayed in the navigation
- *Folder*: Click on *Select* to choose an existing navigation folder. The new folder will be a subfolder of the selected one.
- *Icon*: Add an icon to your folder
- *Order*: Set the position of the folder

In the area *Link*, you can select the folders link target by clicking on the *Select* button and the *Character coding* for the Naviagtion item and its contained navigation items.

3.10.3.1 The *Content* view

Click on the *Content* tab to access this view. Here you can set the *content* of the folder by dropdown menu: *No dynamic contents* or *The folder contains dynamic contents*.

In the first case the process is concluded - you create sub items within the folder with *New> entry*. In the second case, the *Content* area is customized for dynamic contents.

Figure 155 Dynamic contents in folder

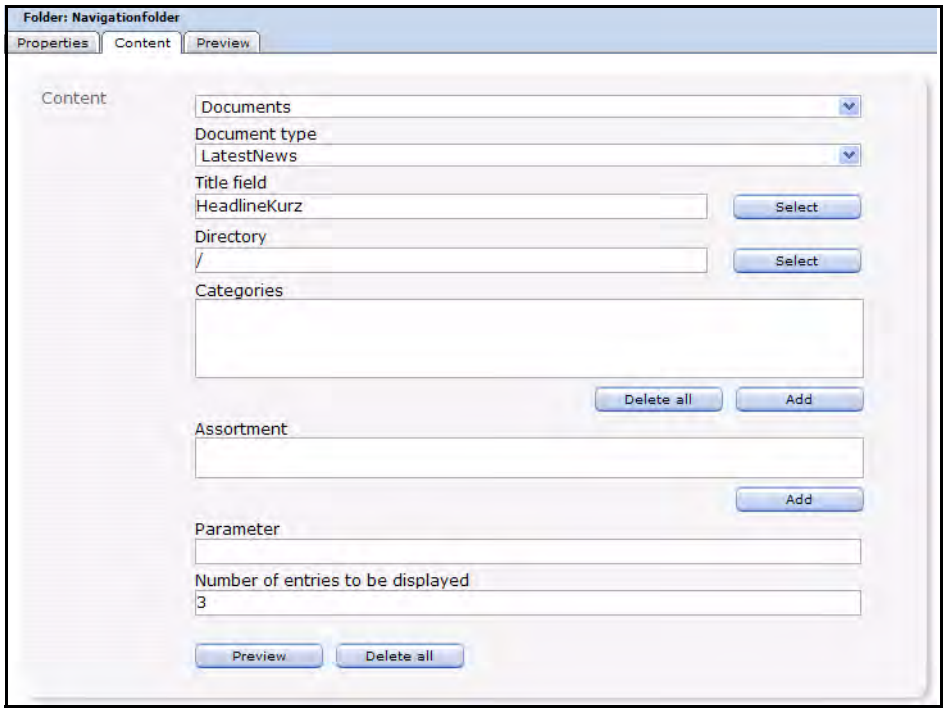


Figure 156 Dynamic navigation folder in the explorer tree

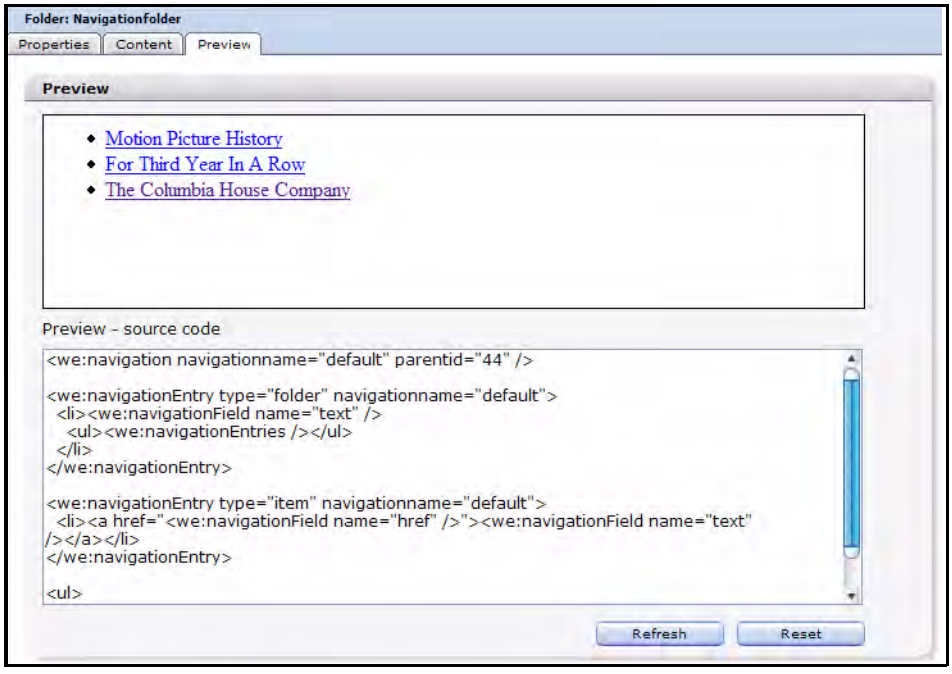
Create now the dynamic folder contents analogous to Section 3.10.2.2, "Dynamic navigation entry,". The new entries are displayed in the explorer bar to the left after saving. With click on the *Preview* button, the entries are displayed with path (see. Figure 152, "Preview of the navigation entries,").

Now to the difference between the dynamic entries generated here in a navigation folder and analogously in comparison to generated entries with *New* > *Entry*. To make it short, the dynamic entries are saved in this folder according to the folder rules statically. This means that, if in the entries (in this example documents of the document type LatestNews) something changes, the folder must be saved in the navigation tool anew. In the comparable example Section 3.10.2.2, "Dynamic navigation entry," a new entry would be automatically added immediately.

3.10.3.2 The *Preview* view

Click on the *Preview* Tab to open this view.

Figure 157 The *Preview* view



In the upper area, the generated navigation entries are displayed, in the lower area the matching source code. You may customize the source code in the lower area and check the changes with a click on the *update* button.

3.11 Rules for the highlighting

With the navigation tool it is possible to highlight the active entry. Besides, the navigation tool automatically recognizes if the active document is located in the navigation. In addition, all dynamic entries are defined as rules for the Highlighting. A dynamic entry in the navigation which should indicate the first three documents of certain document types selects the concerning navigation entry not only for the indicated three documents, but for all documents of this document type as active. Beside this automatic highlighting, other rules can be defined, in addition. These contain, without the number of the documents to be indicated, the same data fields as a dynamic navigation entry.

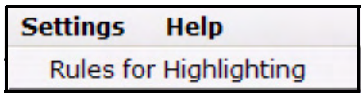
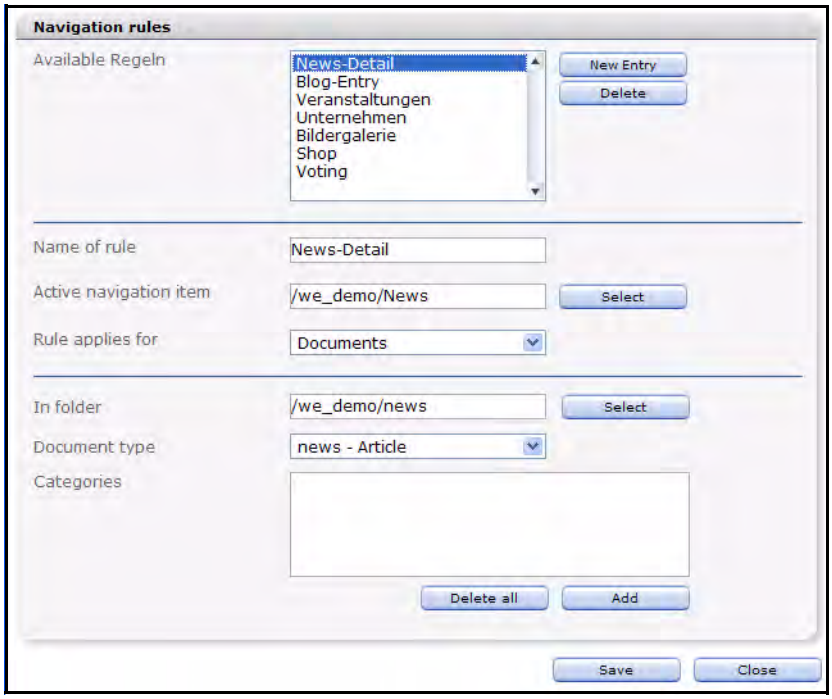


Figure 158 Dropdown *Rules for highlighting*

Click *Preferences* > *Rules for Highlighting* in the navigation tool window to access the menu for the Highlighting rules. Depending on document type and folder, resp. workspace and classes for objects, you can set the navigation entry to be highlighted. (Figure 159 shows an example from the we_demo).

Figure 159 Window *Navigation Rules*



3.12 Tags for the navigation tool

The output of a navigation strongly differs from other elements with repeating character like *we:block* or *we:listview*. After the navigation is initialized, every entry can be formed in the navigation explicitly. Instead of having to access many different if-tags, the appearance of single entries can be customized. Especially the treatment of multi-level navigations is more clear.

3.12.1 we:navigation

we:navigation initializes a navigation. It has the attributes "navigationname" and "parentid". The "name" parameter assigns a name to the navigation, which is used later

to access the navigation. The parameter “parentid” defines the element from which the navigation should be generated. With “parentid=0” the complete navigation is initialized.

```
<we:navigation name="default" parentid="0" />
```

3.12.2 we:navigationEntry

The appearance of a navigation entry is set with we:navigationEntry. Differently than with listview the appearance of an entry of a certain type at a certain level is fixed here explicitly. *we:navigationEntry* has got the parameters *navigationName*, *type*, *level*, *current* and *position*. *navigationName* refers to the name of the navigation defined in *we:navigation*. As *type* "folder" and "entry" may be selected; these may be displayed differently. With *level* you may access a specific level of the navigation. With the parameter *current*, active elements can be displayed differently from inactive ones. With the parameter *position* one can access elements of a hierarchy level of the navigation in a certain position separately. As *position* any number (1,2,3, …) or "odd", "even" or “last” can be used. Only *navigationName* and *type* are mandatory attributes. If no other attributes are used, the appearance for all entries of the type are set.

The appearance of all folders (all levels) of the navigation is defined as follows:

```
<we:navigationEntry type="folder" navigationName="default">
  <li><we:navigationField name="text" />
    <ul><we:navigationEntries /></ul>
  </li>
</we:navigationEntry>
```

Access all active folders (of all levels) with the additional parameter current="true":

```
<we:navigationEntry type="folder" navigationName="default"
current="true">
  <li id="currentFolder"><strong><we:navigationField name="text"
/></strong>
    <ul><we:navigationEntries /></ul>
  </li>
</we:navigationEntry>
```

The appearance of normal navigation entries:

```
<we:navigationEntry type="item" navigationName="default">
  <li><a href="<we:navigationField name="href"
/>"><we:navigationField name="text" /></a></li>
</we:navigationEntry>
```

As well as active navigation entries:

```
<we:navigationEntry type="item" navigationName="default"
current="true">
  <li><a id="currentA" href="<we:navigationField name="href"
/>"><we:navigationField name="text" /></a></li>
</we:navigationEntry>
```

3.12.3 we:navigationEntries

The *we:navigationEntries* tag can only be used within a *we:navigationEntry* tag. It is used as a placeholder for all entries within a navigation folder. *we:navigationEntries* has no parameters.

```
<we:navigationEntry type="folder" navigationName="default"
current="true">
  <li id="currentFolder"><strong><we:navigationField name="text"
/></strong>
    <ul><we:navigationEntries /></ul>
  </li>
```

```
</we:navigationEntry>
```

3.12.4 we:navigationField

we:navigationField can only be used within a *we:navigationEntry*. It is used as a placeholder for a field of a navigation entry. The parameter "name" issues the name of the field to be displayed.

Possible values are:

- text: Entered text
- href: Path to the target document
- icon: Path to the selected icon
- level: Level on which the entry is located
- position: position of the entry
- current: shows if the entry is currently active
- id: id of the navigation entry
- parentid: Id of the parent element of the navigation entry
- type: folder or entry

3.12.5 we:navigationWrite

With *we:navigationWrite*, the navigation is put out; the only parameter is "navigationName", refering to the navigation created with that same name.

```
<ul>
  <we:navigationWrite name="default" />
</ul>
```

3.13 Example

Navigations created with the navigation tool can be designed arbitrarily. In the following example this is made clear with the help of the navigation of the webEdition demo "we_demo". Starting point is a built in navigation and a complete Highlighting set of rules. Here only the application of the we:tags should be explained.

At first the navigation is initialized:

```
<we:navigation parentid="0" name="mainNavigation" />
```

Name of the navigation is `mainNavigation`, as `parentid` 0 is selected: thus, the complete navigation is used.

The navigation to be displayed contains two levels. It should be designed as accessible as possible. Hence, the navigation is displayed as an interlocking unsorted list, all layout information is included in a CSS file (*/we_demo/css/screen/basic.css*). Only the contents of the active folder should be indicated, this should be highlighted at the same time optically. Same goes for the "normal" navigation entries, here,too, the active entry should be highlighted.

At first, the output of the folders is set. For inactive folders, only a link to the start page of the corresponding rubric is put out:

```
<we:navigationEntry navigationName="mainNavigation" type="folder">
  <li>
```

```
    <a href="<we:navigationField name="href" />">
      <we:navigationField name="text" />
    </a>
  </li>
</we:navigationEntry>
```

Active folders should additionally display all sub entries. Additionally, an ID is set. According to that ID, a different appearance is assigned to the active link. An active folder contains a list element with the link as well as all contained elements within another unsorted list.

```
<we:navigationEntry navigationName="mainNavigation" type="folder"
current="true">
  <li id="li_nav1_open">
    <a id="a_nav1_open" href="<we:navigationField name="href" />">
      <we:navigationField name="text" />
    </a>
    <ul><we:navigationEntries /></ul>
  </li>
</we:navigationEntry>
```

Now the standard navigation entries are still absent. These are displayed as a list element.

```
<we:navigationEntry navigationName="mainNavigation" type="item">
  <li>
    <a href="<we:navigationField name="href" />">
      <we:navigationField name="text" />
    </a>
  </li>
</we:navigationEntry>
```

Active elements are provided with certain "id" and are highlighted by the CSS file visually.

```
<we:navigationEntry navigationName="mainNavigation" type="item"
current="true">
  <li id="li_nav2_open">
    <a id="a_nav2_open" href="<we:navigationField name="href" />">
      <we:navigationField name="text" />
    </a>
  </li>
</we:navigationEntry>
```

In the end, the navigation must be displayed. This happens with *we:navigationWrite*. Because the navigation contains an unsorted list, *we:navigationWrite* must be surrounded by an -tag.

```
<ul>
  <we:navigationWrite name="mainNavigation" />
</ul>
```

An extract from the fomatted output of this navigation could look as follows:

```
<ul>
  <li id="li_nav1_open">
    <a id="a_nav1_open" href="/we_demo/accessible/index.php">
      Barrierefreiheit
    </a>
    <ul>
      <li>
        <a
href="/we_demo/accessible/information.html">Information</a>
      </li>
      <li>
        <a href="/we_demo/accessible/demo.html">Diese Demo</a>
```

```
        </li>
      </ul>
    </li>
    <li><a href="/we_demo/news/index.php">News</a></li>
    <li><a href="/we_demo/blog/index.php">Blog</a></li>
    <li><a href="/we_demo/events/index.php">Veranstaltungen</a></li>
    <li><a href="/we_demo/company/index.php">Unternehmen</a></li>
    <li><a href="/we_demo/imprint/index.html">Impressum</a></li>
    <li><a
  href="/we_demo/image_gallery/index.php">Bildergalerie</a></li>
    <li><a href="/we_demo/customer/index.php">Kundenbereich</a></li>
    <li><a href="/we_demo/shop/index.php">Shop</a></li>
    <li><a href="/we_demo/guestbook/index.php">Gästebuch</a></li>
    <li><a href="/we_demo/newsletter/index.php">Newsletter</a></li>
    <li><a href="/we_demo/voting/index.php">Voting</a></li>
  </ul>
```

Figure 160 Navigation we_demo

Thie results, including the CSS, on the we_demo webseite in this output.

4 Creating templates—the role of the administrator

The webEdition content management system differentiates between the role of the editor and the role of the administrator. The following chapter describes how administrators can use webEdition to create templates, as well as some of the features included in webEdition that can assist in this task. An administrator must have a knowledge of HTML and PHP programming to create a template.

4.1 What is a template?

Templates form the framework for a Web page. In a content management system, content is separated from the layout. The layout is created with templates designed by an administrator, while the content is created by editors in documents based on templates. Templates are normally created using HTML and expanded using we:tags.

Note: This manual always refers to the sample pages supplied with the software. Sample pages are located in the we_demo directory within the explorer menu.

4.1.1 What are we:tags?

we:tags comprise a tag language based on XML which contain HTML, JavaScript and PHP functions. They always begin with "we:". All webEdition features are defined by these tags. Like HTML tags, we:tags can have attributes. Although there are we:tags that require an end tag (such as <we:block></we:block>), most do not require an end tag.

4.1.2 How do I use we:tags?

There are many we:tags that serve many functions. All we:tags are described in detail in *The webEdition Tag Reference Manual*, which is available on the webEdition Web site. Some of the most commonly used we:tags are listed in the following table:

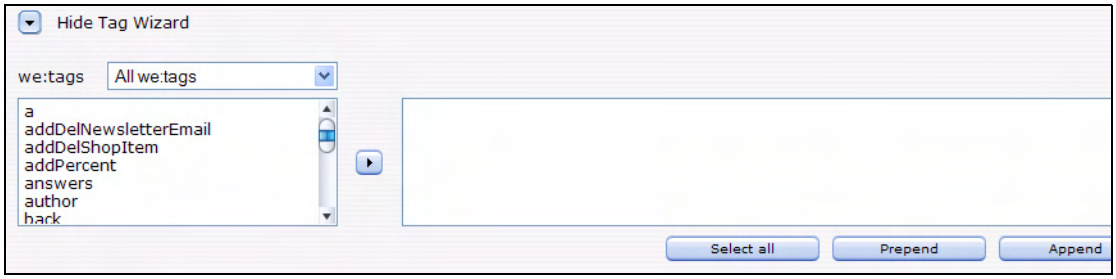
we:tag	Description
we:input (text)	Creates a single input box.
we:input (date)	Creates a drop-down menu to enter date and time.
we:input (checkbox)	Creates a check box.
we:input (choice)	Creates a normal single-line input box with a drop-down menu beside it.
we:textarea	Creates a multi-line input box.

we:tag	Description
we:keywords	Creates input boxes for the meta tags.
we:description	Creates input boxes for the meta tags.
we:title	Creates input boxes for the meta tags.
we:search (textinput)	Creates a single-line input box (also on the finished page) that can be used for queries.
we:search (textarea)	Creates a multi-line input box for queries.
we:select	Creates a select box.
we:category	Assigns the page / site to a category (important for summary pages).
we:linklist	This tag is used to generate expandable link lists.
we:link	Defines the link in link lists.

4.1.2.1 The Tag Wizard

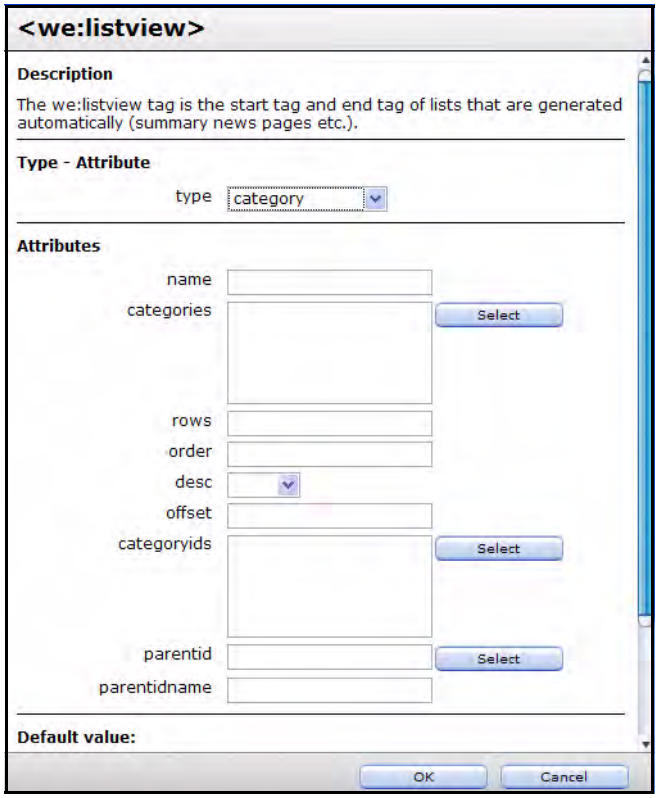
webEdition provides you with a Tag Wizard when you are creating or editing a template so that you do not need to memorize the functions and names of all of the we:tags. (The Tag Wizard can be accessed from the template *Edit* view; see Section 4.2.2, "The Templates Edit view" on page 127). The scroll box on the left contains all available we:tags, listed in alphabetical order (see Figure 161). Select a we:tag and double click or click on the arrow between the two boxes.

Figure 161 The Tag Wizard



If this we:tag allows or requires you to assign attributes to it, a new window will open with a brief description of the tag and fields for all the attributes you can assign to it (see Figure 162). Any attribute you are required to assign are marked with an asterisk (*). The tag wizard displays only the attributes which are appliable to the selected Tag and *type*. In the example, only the viable attributes for <we:listview type=category> are displayed.

Figure 162 Assigning attributes using the Tag Wizard



After saving your attributes, the window closes and the we:tag (complete with the selected attributes and correct syntax) will appear in the Tag Wizard text box to the right of the arrow button. Using copy and paste, insert the we:tag into the source code that appears in the text input box. You can also use the *Select all*, *Prepend* or *Append* buttons to insert the tag at the beginning or the end of the template code.

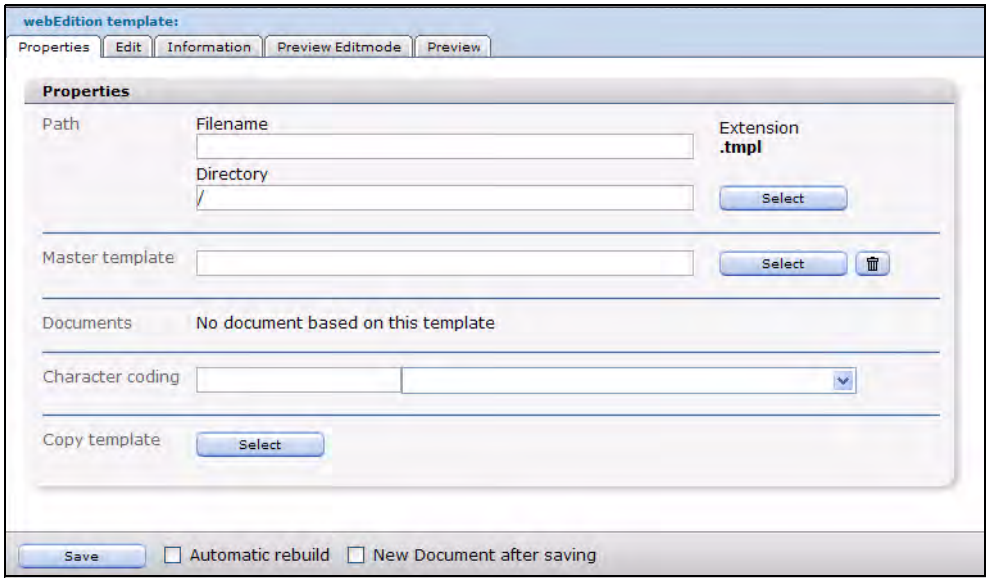
4.2 The five views when working with templates

In Chapter 2, we discussed how the editing screens and the available views are different depending on whether you are working with a document or a template. The following sub-sections describes each of these views as they pertain to your webEdition templates.

4.2.1 The template *Properties* view

You use the *Properties* view to define the name, location of your webEdition template (see Figure 163).

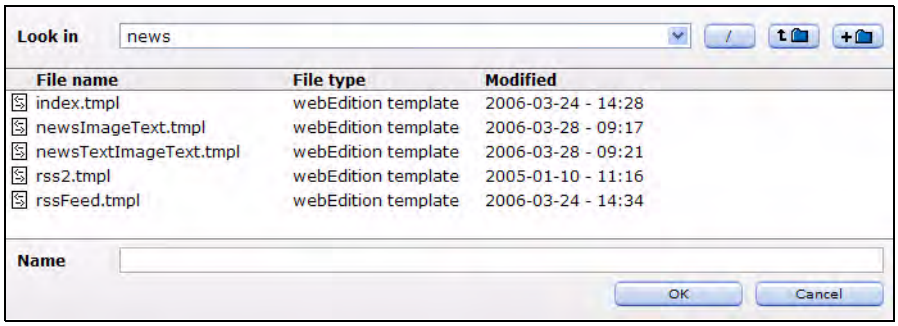
Figure 163 The *Templates Properties* view



The properties view has two areas: *Path* and *Copy template*.

- *Path*. This area allows you to enter the name of the template and the directory in which it is to be saved. Select the directory by clicking the *Select* button and using the file manager. To avoid input errors, the directory cannot be entered by hand. The file extension has been preset to .tmpl.
- *Master template*. Click on the *Select* Button to assign a master template to the template.
- *Documents*. Here, the documents are displayed which are based on this template. Click on one of these documents to directly access the document.
- *Character coding*. Here you can adjust the character coding for the template (e.g., UTF-8 or ISO_8859-1). This encoding is used for documents based on this template.
- *Copy template*. You can also import an existing webEdition template by clicking the select button in the *Copy template* area. Select the template you wish to import from the file manager and click *OK* (see *Figure 164*).

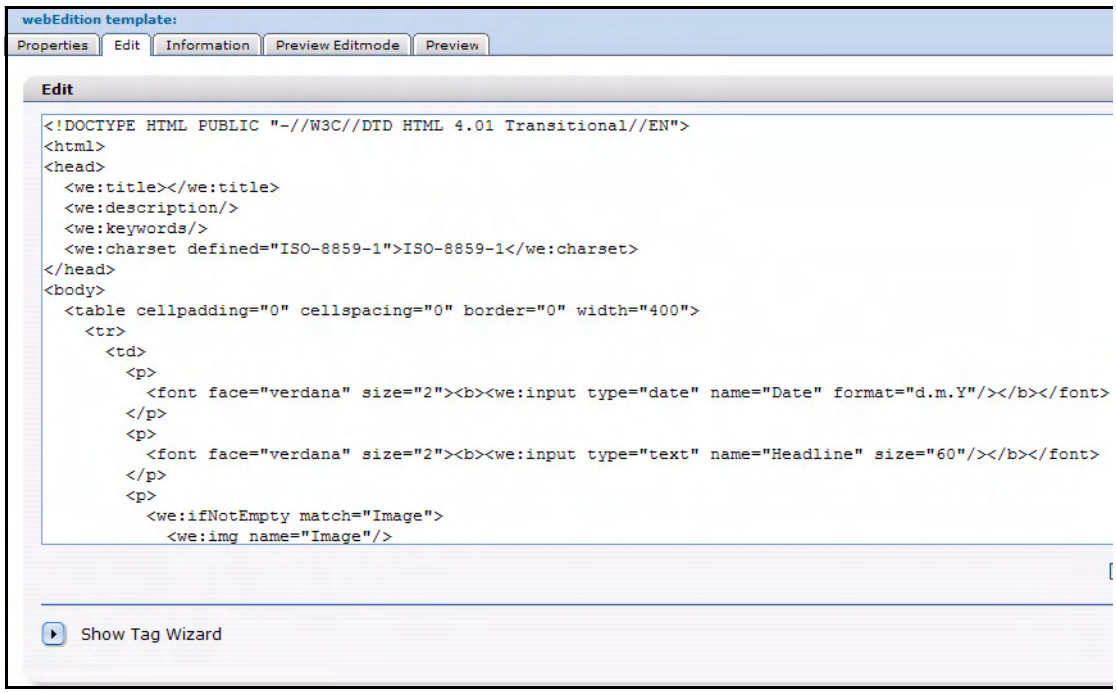
Figure 164 Copying a template using the file manager



4.2.2 The *Templates Edit* view

The *Edit* view displays the HTML code for the template (see Figure 165). The *Edit* view provides a text editing programme in which you can change or expand the standard HTML code and insert various we:tags.

Figure 165 The *Templates Edit* view

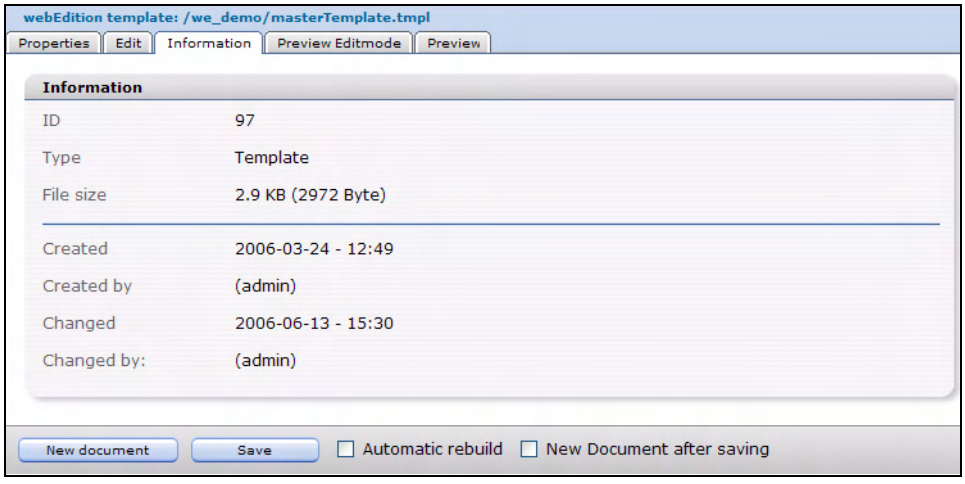


Below the text-editing area, you will find the webEdition Tag Wizard. The wizard assists you in finding and entering the we:tags and assigning the proper attributes to them. For more details on how the we:Tag Wizard works, consult Section 4.1.2.1, "The Tag Wizard" on page 124.

4.2.3 The *Templates Information* view

The *Information* view provides you with general information on the storage path, the size of the template and when it was created or changed (see Figure 166).

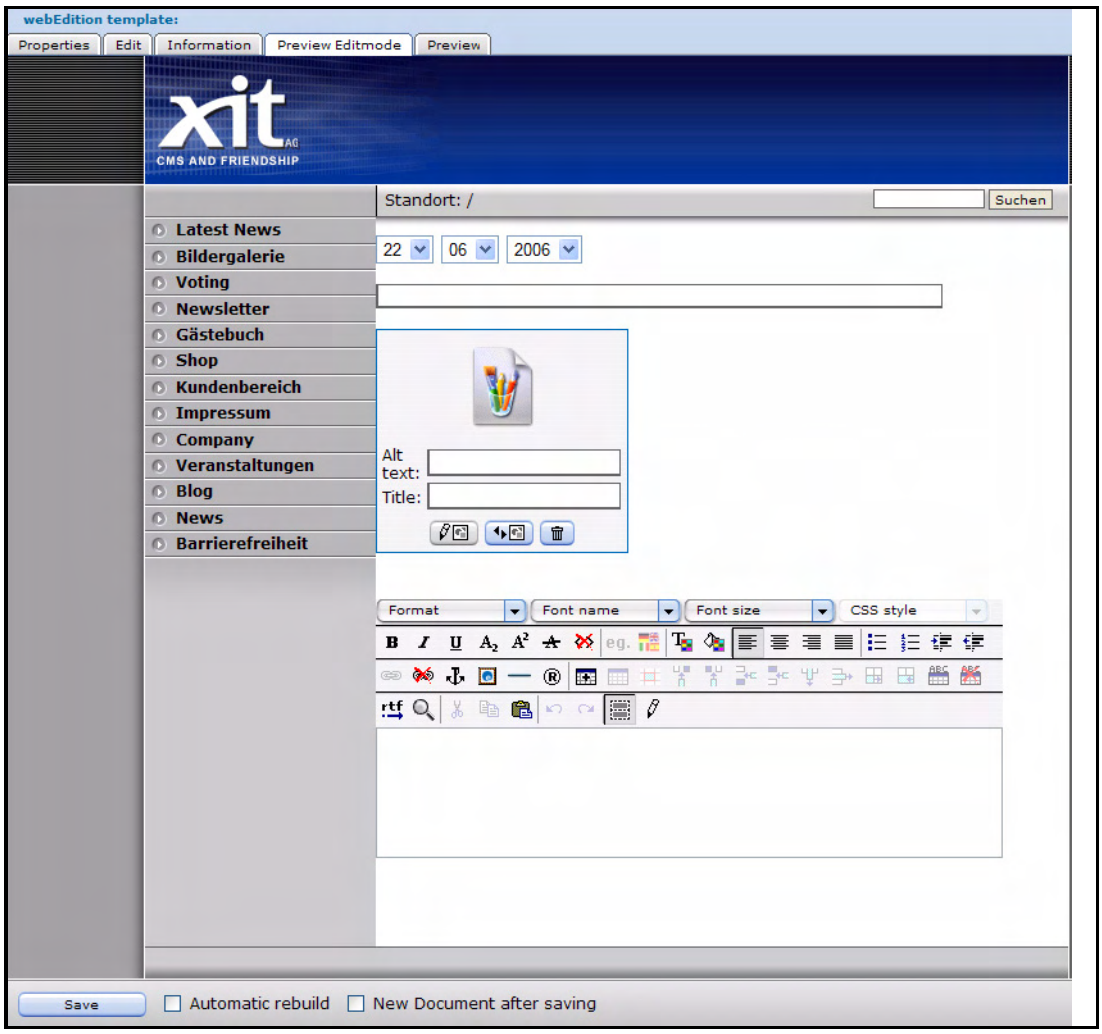
Figure 166 The *Templates Information* view



4.2.4 The *Templates Preview Editmode* view

The *Preview Editmode* view displays the layout of the template as it will appear in edit mode when used by the editor (see Figure 167 - the example shows a default template with the *masterTemplate.tmpl* as Mastertemplate from *we_demo*).

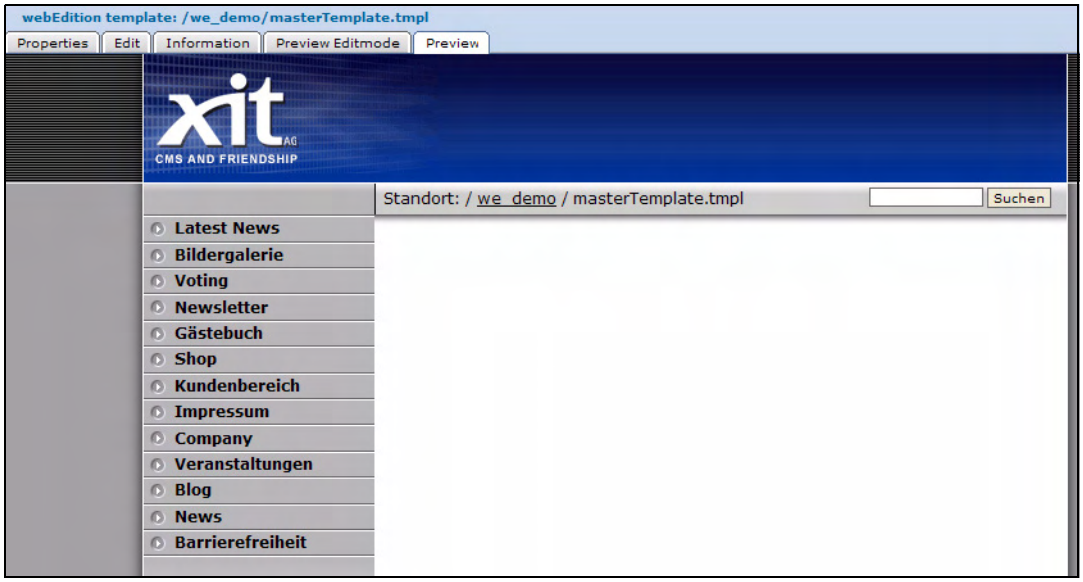
Figure 167 The *Templates Preview Editmode* view



4.2.5 The *Templates Preview* view

The *Preview* view displays how a document based on the template will be displayed to a websitevisitor.

Figure 168 ***Template Preview*** **view**



4.3 Creating Mastertemplates

With a Mastertemplate it is possible to set the layout of your complete web site. It defines the design frame for your the web site. A content template is a template which is "wrapped up" by the main template. The main template is as created as a normal template, nevertheless, it must contain the tag <we:content>. The complete start and end tag, including all enclosed, is replaced later with the code of the content template. However, between the start tag and final tag code can be used which is displayed also in the preview of the templates, so that one has a better image of the document. The <we:content> tag can also be used without end tag.

Example

```
<html>
<head>
<we:title></we:title>
<we:keywords></we:keywords>
<we:description ></we:description>
</head>
<body>
<we:content><b>Example-Code</b> Is displayed in the preview of
the template</we:content>
</body>
</html>
```

or without end tag:

```
<html>
<head>
<we:title></we:title>
<we:keywords></we:keywords>
<we:description ></we:description>
</head>
<body>
<we:content/>
</body>
</html>
```

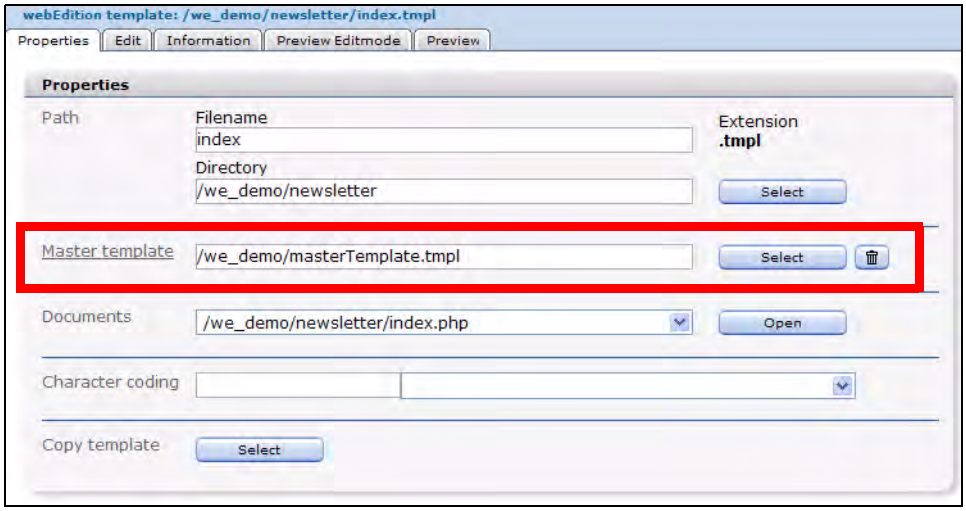
Figure 169 Charting

The Mastertemplate *masterTemplate.tmpl* in we_demo surrounds the whole page.

- 1: The different templates are integrated into the <we:content> area
- 2: The navigation is once integrated with <we:include> into the main template and thus always visible
- 3: Auch der Header ist über ein includiertes Dokument eingebunden, da er ja nicht ständig verändert werden muß

The Mastertemplate can be assigned to templates in the *properties* view of the template (see Section 4.2.1, "The template Properties view,").

Figure 170 *Properties* of a template



When a template with a defined main template is saved, both templates are joined in the process and written in the file system. This joined template is treated like a single template. Hence, a Rebuild about templates and if necessary about the affected documents must also occur if the main template is changed later.

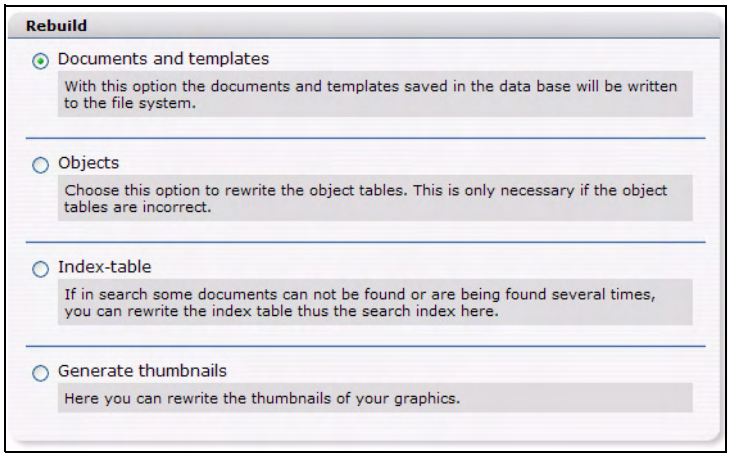
4.3.1 Including templates

If you set the attribute *type="template"* in the <*we:include*> tag (`<we:include type="template" id="templateID">`), a template (with the given ID) is inserted instead of a document.

4.3.2 Template Rebuild

To write anew all templates, a template rebuild can be performed. This is necessary, e.g. if a Mastertemplate was changed and the rebuild process was not performed while saving the template. Select *File* > *Rebuild...*. In the next dialog, select *Documents and templates* and then click on *Next*. Now select *All Templates* and press the *Next* button.

Figure 171 Template Rebuild



4.4 Creating templates

webEdition allows you to create new templates, or to import existing ones.

4.4.1 Creating templates using webEdition

If you wish to use webEdition to create a template, you enter the necessary HTML code in the text/code editing area in the template *Edit* view (see Section 4.2.2, "The Templates Edit view" on page 127).

A template must contain all of the standard HTML tags: <html>, <head> and <body> with their end-tags. In contrast, if you wish to include this template as a document using the <we:include> tag, you do not use these HTML tags.

When linking references (images, links, etc.) that determine the design of several pages (that can be spread out over different directory levels), that is the layout images, you must create an absolute link to the images. The URL is based on the root directory on the Web server (of your host). This is the only way to ensure that a layout image can always be found, no matter where the files using that image have been saved.

Sample template:

```
<html>
<head>
<we:title>CMS Channel</we:title>
<we:description>Demo-Website for the CMS
webEdition</we:description>
<we:keywords>cms,webEdition</we:keywords>
<style media="screen" type="text/css"><!--
#menu1 { position: absolute; z-index: 5; top: 63px; left: 131px;
width: 122px; height: 10px;visibility: hidden  }
#dummy{ position: absolute; z-index: 1; top: 0px; left: 0px; width:
400px; height: 400px;visibility: hidden  }
--></style>
<link href="<we:url id="89" />" rel="styleSheet" type="text/css">
</head>
<body  background="/root/layout_images/bg.gif" bgcolor="white"
leftmargin="0" marginwidth="0" topmargin="8" marginheight="8">
<we:form id="114" method="get"><table border="0" cellpadding="0"
cellspacing="0" width="620">
<tr>
<td width="27"></td>
<td width="50"><img src="/root/layout_images/we_logo.gif"
width="50" height="50" border="0"></td>
<td width="54"></td>
<td><span class="headline"> CMS Channel - </span><span
class="headline_small">News</span></td>
<td class="normal" width="74"><we:date type="js"
format="m/d/Y"/> </td>
```

4.4.2 Creating a new template

Use the following procedure to create a new template.

Procedure 9 Creating a new template

At the webEdition main screen

1 Select *File* > *New* > *Template*.

The webEdition Template Properties view appears.

2 In the *Path* area of the *Properties* view, enter a name for the new template in the Filename field.

3 In the *Path* area of the *Properties* view, choose a directory for the new template by clicking the *Select* button adjacent to the *Directory* field.

a Navigate through the directory menus until you find the location for the new file.

b Click on the name of the target directory, then click *OK*.

4 If you wish to copy an existing template, click on the *Select* button in the *Copy template* area.

a Navigate through the directory menus until you find the location for the new file.

b Click on the name of the target directory, then click *OK*.

5 Click the *Save* button located at the bottom of the screen.

webEdition opens a read-only information box informing you that the new file has been successfully saved.

6 Click the *OK* button.

The newly created file is now listed in the explorer menu and you can now start editing your template, or using it as the basis of a new document.

7 You have completed this procedure.

4.4.3 Saving a template

You can save the template while working in any of the four views by clicking the *Save* button at the bottom of the screen (see Figure 167, "The Templates Preview Editmode view" on page 128).

If you change an existing template that is being used by documents, you will be asked whether you want to re-save the corresponding documents. If you select yes, the template and all documents that use the template will be re-saved. If you select no, only the template will be saved and any documents based on this template will continue to use the old template. The option *Automatic rebuild* causes an automatic re-save of all documents which are based on this template.

4.4.4 Updating/modifying a template

To update or modify a template, either single-click the template in the explorer menu or select your template from the file browser by clicking on *File* > *Open* > *Template...* in the webEdition main screen. Make your changes in the *Edit* view and then save the template.

4.4.5 Deleting a template

You can delete a template by selecting menu item *File* > *Delete* > *Templates*. The application will then switch to delete mode. Use the explorer menu to select the template you wish to delete and click the *OK* button. Confirm or cancel your action.

Note: If a template is being used by one or more documents, you cannot delete it. If you try to do so, an error message will be displayed to this effect. If you still wish to delete the template, you will have to delete or modify all the documents that use the template before you can delete the template itself.

4.4.6 Adopting an existing Web site

4.4.6.1 Adopting an entire page

Determine how many HTML pages your current Web site uses. Group your pages according to their basic layouts in order to identify the pages that share the same layout but differ in their content. You can use these pages to create your new templates.

Procedure 10 Adopting an existing page

In the webEdition main screen

1. Create a new template, by clicking *File* > *New* > *Template*. (see Procedure 9, "Creating a new template" on page 132).
2. Click on the *Edit* tab to access the *Edit* view.
3. Copy the source code from the old HTML page you wish to use as a template into the code/text-editing area in the *Edit* view.
4. Replace the hard-coded HTML tags, text and images with their we:tag equivalents.

HTML tag	we:tag
images (<img src="...>)	<we:img>
Text	<we:textarea>
Heading	<we:input type="text">

Example
To replace an HTML image tag:

Old code:

```
<tr>
<td><img scr="example1.jpg" width="100" height="30"
border="1"></td>
  <td><img src="example2.jpg" width="120" height="30"
border="1"></td>
</tr>
```

New code:

```
<tr>
<td><we:img name="example1"></td>
  <td><we:img name="example2"></td>
</tr>
```

And now instead of seeing images in the template preview, you will see a placeholder.

Example
To replace a text tag:

Old code:

```
<strong>>This is a sample heading in bold and italics</strong>
```

New code:

```
<strong><we:input type="text" name="sample"
size="20"></strong>
```

5. Create a new webEdition page based on the new template.
6. Copy the content from your original page(s) into the input boxes in the template(s) using copy and paste and/or import the images to the database. See Section 3.4, "Working with graphical documents" on page 73.
7. Save each page.

   The new webEdition page is saved in the same location as the old page, thereby overwriting it.

8. You have completed this procedure.

You can find further information about the we:tags, their attributes, and their functions in *The webEditionTag Reference* that is available from the webEdition home page.

4.4.6.2 Adopting portions of a page

If you only want to manage portions of specific pages with webEdition, replace these portions with their we:tag equivalents and then save the page as a template. The template can be used to recreate the page.

4.4.7 Integrating your own we:tags

Now it is possible, to integrate your own we:tags in webEdition. For these we:tags to be correctly recognized by the system, some rules must be considered. If, e.g., the we:tag <*we:b*> is to be created, the following steps have to be executed:

Procedure 11 Create your own we:tag

1 Create the php-function: For *we:b* the file */webEdition/we/include/we_tags/custom_tags/we_tag_b.inc.php* with the function

```
function we_tag_b($attribs, $content) {
  ... php - code ...
  return $text;
}
```

has to be created. This may comprise of any php-code.

$attribs
contains an associative Array with attribute name as key and attribute value as a value

$content
contains the contents between opening and closing we:tag

php-code
is the placeholder for your own functional code

2 Creation of the information file for the Tag Wizard: The information file for your own we:tags which the Tag Wizard accesses is located in the directory */webEdition/we/include/weTagWizard/we_tags/custom_tags*. For <*we:b*> the file *we_tag_b.inc.php* has to be created.

3 For the required attributes it is obvious to copy an existing we:tag and to customize the attribute names / attribute types. If your self-created we:tag is correctly put in, it will be displayed both in the list of all we:tags on the *edit* area of a template and in the Tag Wizard.

4.4.8 Creating templates from documents made in HTML editing programs

You can create templates using an HTML editing program (such as Macromedia Dreamweaver or Adobe GoLive). For more information, see Section 4.4.6.1, "Adopting an entire page" on page 133.

Procedure 12 Creating a template from an HTML editor

Begin in your preferred HTML editor

1 Create a page in your preferred editing program.

2 Move all the layout images you need for the page to a folder called *images*.

3 Upload the images via FTP to your Web server in the directory you have chosen (such as layout_images in the root directory).

4 On your web server, use the global search and replace feature to replace all images/ with /images/.

5 Use the copy and paste feature to copy the source code for the HTML page to the webEdition template *Edit* view (*See Templates* > *Edit* view, upper text box).

6 Replace the text, images, Flash movies etc. that you wish to edit with the appropriate we:tags.

7 Define the functions you wish to include in the page using the appropriate we:tags.

8 Save the template.

9 You have completed this procedure.

4.5 Tags for the navigation tool

The output of a navigation strongly differs from other elements with repeating character like *we:block* or *we:listview*. After the navigation is initialized, every entry can be formed in the navigation explicitly. Instead of having to access many different if-tags, the appearance of single entries can be customized. Especially the treatment of multi-level navigations is more clear.

4.5.1 we:navigation

we:navigation initializes a navigation. It has the attributes "name" and "parentid". The "name" parameter assigns a name to the navigation, which is used later to access the navigation. The parameter “parentid” defines the element from which the navigation should be generated. With “parentid=0” the complete navigation is initialized.

```
<we:navigation name="default" parentid="0" />
```

4.5.2 we:navigationEntry

The appearance of a navigation entry is set with we:navigationEntry. Differently than with listview the appearance of an entry of a certain type at a certain level is fixed here explicitly. *we:navigationEntry* has got the parameters *navigationName*, *type*, *level*, *current* and *position*. *navigationName* refers to the name of the navigation defined in *we:navigation*. As *type* "folder" and "entry" may be selected; these may be displayed differently. With *level* you may access a specific level of the navigation. With the parameter *current*, active elements can be displayed differently from inactive ones. With the parameter *position* one can access elements of a hierarchy level of the navigation in a certain position separately. As *position* any number (1,2,3, …) or "odd", "even" or “last” can be used. Only *navigationName* and *type* are mandatory attributes. If no other attributes are used, the appearance for all entries of the type are set.

The appearance of all folders (all levels) of the navigation is defined as follows:

```
<we:navigationEntry type="folder" navigationName="default">
  <li><we:navigationField name="text" />
    <ul><we:navigationEntries /></ul>
  </li>
</we:navigationEntry>
```

Access all active folders (of all levels) with the additional parameter current="true":

```
<we:navigationEntry type="folder" navigationName="default"
current="true">
  <li id="currentFolder"><strong><we:navigationField name="text"
/></strong>
    <ul><we:navigationEntries /></ul>
  </li>
</we:navigationEntry>
```

The appearance of normal navigation entries:

```
<we:navigationEntry type="item" navigationName="default">
  <li><a href="<we:navigationField name="href"
/>"><we:navigationField name="text" /></a></li>
</we:navigationEntry>
```

As well as active navigation entries:

```
<we:navigationEntry type="item" navigationName="default"
current="true">
  <li><a id="currentA" href="<we:navigationField name="href"
/>"><we:navigationField name="text" /></a></li>
</we:navigationEntry>
```

4.5.3 we:navigationEntries

The *we:navigationEntries* tag can only be used within a *we:navigationEntry* tag. It is used as a placeholder for all entries within a navigation folder. *we:navigationEntries* has no parameters.

```
<we:navigationEntry type="folder" navigationName="default"
current="true">
  <li id="currentFolder"><strong><we:navigationField name="text"
/></strong>
    <ul><we:navigationEntries /></ul>
  </li>
</we:navigationEntry>
```

4.5.4 we:navigationField

we:navigationField can only be used within a *we:navigationEntry*. It is used as a placeholder for a field of a navigation entry. The parameter "name" issues the name of the field to be displayed.

Possible values are:

- text: Entered text
- href: Path to the target document
- icon: Path to the selected icon
- level: Level on which the entry is located
- position: position of the entry
- current: shows if the entry is currently active
- id: id of the navigation entry
- parentid: Id of the parent element of the navigation entry
- type: folder or entry

4.5.5 we:navigationWrite

With *we:navigationWrite*, the navigation is put out; the only parameter is "navigationName", refering to the navigation created with that same name.

```
<ul>
  <we:navigationWrite name="default" />
</ul>
```

4.5.6 Example

Navigations created with the navigation tool can be designed arbitrarily. In the following example this is made clear with the help of the navigation of the webEdition demo "we_demo". Starting point is a built in navigation and a complete Highlighting set of rules. Here only the application of the we:tags should be explained.

At first the navigation is initialized:

```
<we:navigation parentid="0" name="mainNavigation" />
```

Name of the navigation is `mainNavigation`, as `parentid` 0 is selected: thus, the complete navigation is used.

The navigation to be displayed contains two levels. It should be designed as accessible as possible. Hence, the navigation is displayed as an interlocking unsorted list, all layout information is included in a CSS file (*/we_demo/css/screen/basic.css*). Only the contents of the active folder should be indicated, this should be highlighted at the same time optically. Same goes for the "normal" navigation entries, here,too, the active entry should be highlighted.

At first, the output of the folders is set. For inactive folders, only a link to the start page of the corresponding rubric is put out:

```
<we:navigationEntry navigationName="mainNavigation" type="folder">
  <li>
    <a href="<we:navigationField name="href" />">
      <we:navigationField name="text" />
    </a>
  </li>
</we:navigationEntry>
```

Active folders should additionally display all sub entries. Additionally, an ID is set. According to that ID, a different appearance is assigned to the active link. An active folder contains a list element with the link as well as all contained elements within another unsorted list.

```
<we:navigationEntry navigationName="mainNavigation" type="folder"
current="true">
  <li id="li_nav1_open">
    <a id="a_nav1_open" href="<we:navigationField name="href" />">
      <we:navigationField name="text" />
    </a>
    <ul><we:navigationEntries /></ul>
  </li>
</we:navigationEntry>
```

Now the standard navigation entries are still absent. These are displayed as a list element.

```
<we:navigationEntry navigationName="mainNavigation" type="item">
  <li>
    <a href="<we:navigationField name="href" />">
      <we:navigationField name="text" />
    </a>
  </li>
</we:navigationEntry>
```

Active elements are provided with certain "id" and are highlighted by the CSS file visually.

```
<we:navigationEntry navigationName="mainNavigation" type="item"
current="true">
  <li id="li_nav2_open">
    <a id="a_nav2_open" href="<we:navigationField name="href" />">
      <we:navigationField name="text" />
    </a>
  </li>
</we:navigationEntry>
```

In the end, the navigation must be displayed. This happens with *we:navigationWrite*. Because the navigation contains an unsorted list, *we:navigationWrite* must be surrounded by an -tag.

```
<ul>
  <we:navigationWrite name="mainNavigation" />
</ul>
```

An extract from the fomatted output of this navigation could look as follows:

```
<ul>
  <li id="li_nav1_open">
    <a id="a_nav1_open" href="/we_demo/accessible/index.php">
      Barrierefreiheit
    </a>
    <ul>
      <li>
        <a
href="/we_demo/accessible/information.html">Information</a>
      </li>
      <li>
        <a href="/we_demo/accessible/demo.html">Diese Demo</a>
      </li>
    </ul>
  </li>
  <li><a href="/we_demo/news/index.php">News</a></li>
  <li><a href="/we_demo/blog/index.php">Blog</a></li>
  <li><a href="/we_demo/events/index.php">Veranstaltungen</a></li>
  <li><a href="/we_demo/company/index.php">Unternehmen</a></li>
  <li><a href="/we_demo/imprint/index.html">Impressum</a></li>
  <li><a
href="/we_demo/image_gallery/index.php">Bildergalerie</a></li>
  <li><a href="/we_demo/customer/index.php">Kundenbereich</a></li>
  <li><a href="/we_demo/shop/index.php">Shop</a></li>
  <li><a href="/we_demo/guestbook/index.php">Gästebuch</a></li>
  <li><a href="/we_demo/newsletter/index.php">Newsletter</a></li>
  <li><a href="/we_demo/voting/index.php">Voting</a></li>
</ul>
```

This results, including the CSS, on the we_demo webseite in the navigation output.

4.6 Features for developing webEdition templates

The sub sections that follow describe the features that are available in webEdition to assist in the development of templates.

4.6.1 Link lists (<we:link> and <we:linklist>)

The <we:linklist> tag allow you to include expandable and editable link lists in your templates. You can change the appearance of these link lists using HTML commands. For example, you can design the link to be flanked by non-linked text or images. You can create both image links and text links. You can use almost all tags within a link list. To add a link to the linklist, use the <we:link> tag.

When using your template, the editor will be able to edit any link by clicking the *Edit* button or delete a link using the trash can icon. The editor can also add a link at the very end of the link list or can place it before any existing link. See Section 3.7.7, "Link list (<we:linklist>)" on page 102 for examples.

You can also use the <we:prelink> and <we:postlink> we:tags to ensure that all the characters or commands between <we:postlink> and </we:postlink> or those between <we:prelink> and </we:prelink> will not be displayed in the first or last instance. For example, if you separate the links with a <p>, then you want the <p> to appear only

between the links and not directly before or after the link list. To keep this from happening, you would have webEdition skip the first instance of <p>, for example:

```
<we:linklist name="linklist">
  <we:prelink><p></we:prelink>
  <we:link/>
</we:linklist>
```

would produce for the first instance:

```
<a href="link">Linktext</a>
```

for two instances:

```
<a href="link">Linktext</a><p><a href="link">Linktext</a>
```

without the prelink tag, one instance would produce the following:

```
<p><a href="link">Linktext</a>
```

and for two instances:

```
<p><a href="link">Linktext</a><p><a href="link">Linktext</a>
```

Here is an example from a sample Web site:

```
<we:linklist name="Linklist"/>
<tr><td class="link" valign="top"><nobr><we:link></nobr></td>
<td><img src="/root/layout_images/pixel.gif" width="20" height="2"
border="0"></td>
<td valign="top" class="normal"><we:textarea name="Erklaerung"
rows="2"
cols="40"/></td></tr>
<we:postlink>
<tr>
<td colspan="3"><img src="/root/layout_images/pixel.gif" width="2"
height="6" border="0"></td>
</tr>
</we:postlink>
</we:linklist>
```

This code generates the following output:

Figure 172 Example of output for a link list

To allocate a stylesheet to the we:link tag, do this:

```
<we:link name="MyLink" class="foo"/>
```

4.6.2 Summary pages/areas

Summary pages provide an overview of all pages or areas that belong to a document type and/or category. (For example, all “news” or all news belonging to the “Sports” category). In addition, summaries can provide an overview of an entire Web site. Summary pages are also used to generate a results page for searches.

4.6.2.1 Creating a summary page

You use the <we:listview> we:tag to create summary pages. This tag must enclose all the other we:tags needed for summary pages (start-tag and end-tag). The <we:listview>

tag must contain a <we:repeat> tag that defines the appearance of the entries listed in the summary. Everything bracketed by the we:repeat tags—including normal HTML (such as lines in a table) or the we:tag <we:field/>, which displays the content of a given field— will be repeated.

For example, assume that you have defined input boxes <we:input type="text" name="Header"/> and <we:input type="date" name="Date"/> in a news template belonging to the "newsArticle" document type. To generate a summary in which every line contains the date, header and a link to the news page, you would require the following code:

```
<we:listview doctype="newsArticle" rows="10">
  <we:repeat>
    <we:field type="date" name="Date"/>
    <we:field type="text" name="Header">
  </we:repeat>
  <we:back>back</we:back> <we:next>next</we:next>
</we:listview>
```

The <we:back> and the <we:next> tags, which create an automatic link to the previous/next page for entries listed within the summary, can also be placed within <we:listview>, although they cannot be placed within a <we:repeat> tag. However, this is only possible if the number of entries has been limited by the *rows* attribute in <we:listview>. If this attribute has not been specified, all the pages found will be displayed. If the *rows* attribute has been set to 10, only 10 entries will be displayed. Any <we:next> tag that has been included will create another link.

Here is an example from a sample Web site:

```
<we:listview rows="6" doctype="newsArticle" order="Date"
desc="true">
<table border="0" cellpadding="0" cellspacing="0" width="500">
<we:repeat>
<tr>
<td class="normal"><we:field type="date" name="Date"
format="m/d/Y"/></td>
<td><img src="/root/layout_images/pixel.gif" width="10" height="2"
border="0"></td>
<td class="normal"><b><we:field type="text" name="Headline"
hyperlink="on"/></b></td>
</tr>
<tr>
<td><img src="/root/layout_images/pixel.gif" width="90" height="6"
border="0"></td>
<td><img src="/root/layout_images/pixel.gif" width="10" height="6"
border="0"></td>
<td><img src="/root/layout_images/pixel.gif" width="400"
height="6" border="0"></td>
</tr>
</we:repeat>
<we:ifNotFound><tr>
<td colspan="3" class="normal">No News available!</td>
</tr>
<we:else/><tr>
<td colspan="3" class="normal">
<table cellpadding="0" border="0" cellspacing="0" width="100%">
<tr>
<td colspan="2"><img src="/root/layout_images/pixel.gif"
width="10" height="6" border="0"></td>
</tr>
<tr>
```

```
<td class="normal"><we:ifBack><we:back>&lt;&lt;
Back</we:back></we:ifBack></td>
<td class="normal" align="right"><we:ifNext><we:next>weiter
&gt;&gt;</we:next></we:ifNext></td>
</tr>
</table>
</td>
</tr>
</we:ifNotFound>
</table>
</we:listview>
```

Figure 173 shows the output from this code:

Figure 173 Sample summary page

04/01/2003	webEdition Software GmbH
09/28/2001	Time flies and time is running out under stress!
09/27/2001	Talai - the island of dreams!
09/24/2001	Get wind of the new energy!

4.6.2.2 Creating a summary of all your pages

Summaries of all the pages in a Web site are generally needed for search results pages. These summaries are also useful if you have a small Web site with only a few pages and would like to create an automatic navigation facility.

If you want to create a summary of all your pages, do not specify any conditions in <we:listview>. Make sure that either all the pages have the same field names (such as header and date) or that an alternative field has been specified by means of the *alt* attribute in the <we:field> tag. For example: <field name="Title" alt="Headline"/>. In this case, if the title field in the *Property* view is empty, the content of the *Headline* field will be displayed.

There are two reserved field names that you can use: we_path and we_text. we_path will cause the path of the page to be displayed, while we_text will cause the text content of the page to be displayed. You can limit the text content using the max attribute. If no entries have been made in any of the fields, or if the template does not have any input boxes, we_text will not produce any results.

Note: The syntax is we_path, *not* we:path; we_text, *not* we:text.

Here is an example from a sample Web site:

```
<we:listview rows="6">
<table border="0" cellpadding="0" cellspacing="0" width="500">
<we:repeat>
<tr><td class="normal"><b><we:field name="Title" alt="we_path"
hyperlink="on"/></b><br><we:field name="Description" alt="we_text"
max="200"/></td>
</tr>
<tr><td><img src="/root/layout_images/pixel.gif" width="500"
height="6"
border="0"></td>
</tr>
</we:repeat>
<we:ifNotFound>
<tr><td colspan="3" class="normal">Leider nichts gefunden!</td>
```

```
</tr>
<we:else/>
<tr>
<td colspan="3" class="normal">
<table cellpadding="0" border="0" cellspacing="0" width="100%">
<tr><td colspan="2"><img src="/root/layout_images/pixel.gif"
width="10"
height="6" border="0"></td>
</tr>
<tr><td class="normal"><we:ifBack><we:back>&lt;&lt;
Back</we:back></we:ifBack></td>
<td class="normal" align="right"><we:ifNext><we:next>Next
&gt;&gt;</we:next></we:ifNext></td>
</tr>
</table>
</td>
</tr>
</we:ifNotFound>
</table>
</we:listview>
```

Figure 174 shows the output from this code:

Figure 174 Sample of all summary pages

Calling up a summary page using this feature will display all relevant pages. The entry consists of the title and the description of the page (for webEdition pages, the latter is listed in *Properties*). If the page does not have a title or if it is an HTML page, the path will be displayed (alt="we_path") instead of the title. Likewise, if the page does not have a description, the first 200 characters of content on the page will be displayed.

4.6.2.3 Creating a summary of a specified area

If you wish to display a summary of a specified area, you must limit the output in the <we:listview> tag either by specifying the *doctype* attribute or the *categories* attribute.

The following code, for example, displays only those pages that belong to the "News" document type and the "Sports" and "Politics" categories.

```
<we:listview doctype="News" categories="Sport,Politics">
```

4.6.2.4 Using summaries to archive pages

You can use the *offset* attribute as an archiving feature for your news. First, create two summary pages. The <we:listview> tag on the first summary page (current news) might look something like this:

```
<we:listview doctype="News" order="Date" desc="true" rows="10">
```

This page displays the ten most recent news pages, assuming, of course, that your news articles all contain a date field. Don't add the <we:next> and <we:back> tags on this first page. Instead, insert a link to the second summary page (the archive page). The <we:listview> tag on the second summary page will also contain the offset attribute. The <we:listview> will then look something like this:

```
<we:listview doctype="News" order="Date" desc="true" rows="10"
offset="10">
```

Next, insert the <we:next> and the <we:back> tags on this second page. The offset attribute ensures that the first entry listed is, in reality, the tenth news entry. This method allows you to keep the ten most recent news items on the first summary page and to archive the older items on the second summary page. Here is a sample first page (current news) displaying the ten most recent news items:

```
<we:listview doctype="News" rows="10" order="Date" desc="true">
  <we:repeat>
    <we:field type="date" name="Date"/>
    <we:field type="text" name="Headline" hyperlink="true"/><br>
  </we:repeat>
</we:listview>
```

Here is a sample second page (archive) listing all the other news items (beginning with the tenth news item):

```
<we:listview doctype="News" order="Date" desc="true" offset="10">
  <we:repeat>
    <we:field type="date" name="Date"/>
    <we:field type="text" name="Headline" hyperlink="true"/><br>
  </we:repeat>
</we:listview>
```

4.6.3 Creating a photo album

Before the editor of your Web Page can upload images to a photo album, you must create a photo album template in which the thumbnails can be displayed.

A photo album consists of summaries of thumbnails of photos (see Section 3.4.6, "The images Thumbnails view: thumbnail formatting" on page 80). If a visitor to your page clicks on a thumbnail, an enlarged image is displayed either in a separate popup window or in the same window.

A summary is created with the `<we:listview>` tag:

```
<we:listview contenttypes="img" searchable="false" rows="5"
cols="5">
  <we:ifBack><we:back>back</we:back>  </we:ifBack>
  <we:ifNext><we:next>next</we:next></we:ifNext><br>
  <table border="2">
    <we:repeat>
      <we:tr>
        <td>
          <we:field name="Title"><br>
       <we:field border="0" thumbnail="100x70" id="505" type="img"
hyperlink="true" winprops="width=900,height=650,left=-1,top=-1">
          <br><br>
```

```
        </td>
      </we:tr>
    </we:repeat>
  </table>
</we:listview>
```

Use the attribute `contenttypes` to indicate which kind of document should be displayed. You can use the following values:
"img" = images
"wepage" = webEdition documents
"binary" = other documents

Use the attribute `searchable` to define which documents are to be displayed. If you wish only to display documents which are labeled as searchable, set the the value to `searchable="true"`. If you wish to display documents which are not labeled as searchable, set the value to `searchable="false"`.

To display a HTML table with a listview, the attribute `cols` and the tag `<we:tr>` are available. If "`cols`" is set, `rows * cols` entries are displayed. The tag `<we:tr>` creates the HTML tag <tr> at the beginning of a new row; the tag `</we:tr>` creates the HTML tag <tr> at the end of a row.

You can access the meta information of an image (title, description, key words) with the tag `<we:field>` in a listview:

```
<we:field name="Title" />
<we:field name="Description" />
<we:field name="Keywords" />
```

With the attribute `winprops` you can define the properties of the photo that is to appear in a new popup window. (The syntax is equivalent to the *windowFeatures* of *window.open* in Javascript.) To center the window set a value of `-1` for both `left` and `top`.

Example

```
winprops="width=900,height=650,left=-1,top=-1"
```

To display the image, the tag `<we:field>` with the attribute `type="img"` can be used (among other tags), but only without indicating a name. If you want to display a thumbnail instead of the original image, indicate the name of the thumbnail at the attribute `thumbnail`. If `hyperlink="true"` is set, the field will be linked as always. Usually it is linked directly to the image. If that is not preferable, you can set the attribute `id`. Here you enter the ID of a webEdition document. This webEdition document is used for the detailed view of the image. The image will be displayed with a listview with `rows="1"` in the template of such a webEditon document. All necessary settings (such as "offset", "name" of the listview, "workspaceID", and "contenttype"), will be transfered to the document and thus to the listview automatically. The code of the listview appears thus:

```
<we:listview rows="1">
  <we:ifBack><we:back>back</we:back></we:ifBack>  
  <we:ifNext><we:next>next</we:next></we:ifNext>
  <p>
    <we:repeat>
      <we:field name="Title"><br>
      <we:field type="img"><br><br>
    </we:repeat>
```

```
  </p>
</we:listview>
```

4.6.3.1 Creating multiple photo albums

With the tag <we:href> you can allow editors to choose between different photo albums or to create new albums. Each photo album requires its own directory. The editors can put all their photos into their respective directories. Below is a sample of the code:

```
<we:ifEditmode>
  Album:<br>
  <we:href reload="true" directory="true" file="false" type="int"
name="album" output="id">
  <br>
</we:ifEditmode>
<we:setVar to="global" nameto="album" from="top"
namefrom="album_we_jkhdsf_intID" />
<we:listview contenttypes="img" searchable="false" rows="5"
cols="5" workspaceID="\$album">
```

In this example the value which is choosen in `<we:href>` is written in a global value with `<we:setVar>`. This value is then transfered to the listview `i`, attribute `workspaceID`. If `reload="true"` is indicated at the tag `<we:href>`, the page will be reloaded after choosing the directory. If you set the attributes `directory="true"` and `file="false"` directories, not files, will be choosen.

If the attribute `useparent="true"` is used in the detailed view at `<we:ifBack>` and `<we:ifNext>`, it can only be scrolled to the last entry of the summary listview. Example: If image 11 to 20 are displayed in the summary listview and you have chosen an image, you can only scroll forward to entry 20 and back to entry 11.

```
<we:listview rows="1">
  <we:ifBack
useparent="true"><we:back>back</we:back></we:ifBack>  
  <we:ifNext
useparent="true"><we:next>forward</we:next></we:ifNext>
  <p>
    <we:repeat>
      <we:field name="Title"><br><we:field type="img">
      <br><br>
    </we:repeat>
  </p>
</we:listview>
```

4.6.4 Producing and changing pages and objects from the front end

webEdition offers tags which enable the visitor of a Web site to produce and change webEdition documents and objects. With webEdition, one can easily produce such things as a guestbook, or some other type of page in which visitor input is solicited. The <we:form> tag serves as the foundation of this feature.

The <we:form> tag has been extended with two new types: <we:form type="document"> and <we:form type="object"> which can be used depending upon whether one wishes to produce a webEdition document or object. For webEdition documents, the document type must be entered as a value of the attribute *doctype*; for objects, the ID of the class must be entered as a value of the attribute "classid".

You cannot produce or change webEdition documents without the document type. The individual fields are depicted through the <we:userInput/> tag between the start and the end tag of the <we:form>. The value which one allocates to the attribute *type*, will

determine which type it handles. For a full description of these tags and their attributes, see the *webEdition Tag Reference*.

```
<we:form type="document" id="56" doctype="Guestbookentry">
  <table width="500" border="0" cellspacing="0" cellpadding="5"
align="center">
    <tr>
      <td class="t4"><strong>Guestbook entry</strong></td>
      <td align="right" nowrap>
        <img src="/images/pixel.gif" width="1" height="1">
       </td>
     </tr>
     <tr>
       <td class="t3" colspan="2">
         <p>
          <strong>Your name:</strong><br>
          <we:UserInput name="Name" type="text" size="60"/>
         </p>
         <p>
           <strong>Your address:</strong><br>
           <we:UserInput name="Ort" type="text" size="60"/>
        </p><we:userInput name="Datum" type="date" hidden="true"/>
        </td>
      </tr>
      <tr>
        <td class="t3" colspan="2">
          <strong>Your comments:</strong><br>
          <we:userInput type="textarea" name="Text" width="500"
height="200" autobr="true" wysiwyg="true"
commands="prop,list,link,color"/></td>
      </tr>
      <tr>
        <td class="t3" colspan="2"><input type="submit"
value="Submit"> <input type="reset" value="Reset">
      </td>
    </tr>
  </table>
</we:form>
```

For the attribute *id* of the <we:form> tag, the ID of the webEdition document is entered as a value, where the form is to be sent. This can be a connecting page on which one shows the input again, or similarly a page that saves and publishes the input if applicable. Two possibilities are described below:

- The <we:write> tag is placed at the very beginning of the template in order to save the data transferred from the form. For example:

  ```
  <we:write type="document" publish="true"
  doctype="Guestbookentry"/>
  ```

  After the <we:write> tag one can detect with the <we:ifWritten> or <we:ifNotWritten> tags, whether or not the write-process was successful for the webEdition document or object. For example:

  ```
  <we:ifWritten type="document">
  Thank you for the entry in our guestbook!
  <we:else/>
  Unfortunately there is an error in the entry in our guestbook!
  </we:ifWritten>
  ```

- If you would like to send the given data first to a connecting page, use the <we:form> tag in combination with the <we:userInput> tag once again. In contrast to the input-page, one now sets the entry field to "non-editable". You can cause this

to happen by setting the "editable" attribute in the <we:userInput/> tag to the value "false".

For example:

```
<we:form type="document" id="57" doctype="Guestbookentry">
  <p>Enter your comments here:</p>
  <table width="500" border="0" cellspacing="0" cellpadding="5"
align="center">
    <tr>
      <td class="t4"><strong>Guestbook entry</strong></td>
      <td align="right" nowrap>
        <img src="/images/pixel.gif" width="1" height="1">
      </td>
    </tr>
    <tr>
      <td class="t3" colspan="2">
        <p>
          <strong>Your name:</strong><br>
        <we:UserInput name="Name" type="text" editable="false"/>
        </p>
        <p>
          <strong>Your address:</strong><br>
          <we:UserInput name="address" type="text"
editable="false"/>
        </p>
        <we:userInput name="Datum" type="date" hidden="true"/>
      </td>
    </tr>
    <tr>
      <td class="t3" colspan="2">
        <strong>Your comments:</strong><br>
        <we:userInput type="textarea" name="Text"
editable="true"/>
      </td>
    </tr>
    <tr>
      <td class="t3" colspan="2">
        <input type="button" value="Back"
onclick="history.back()"> 
        <input type="submit" value="Submit"> 
        <input type="reset" value="Reset">
      </td>
    </tr>
  </table>
</we:form>
```

ATTENTION

If you give a value to the attribute "name" in the <we:form> tag, you must also enter the name of the form as a value of the attribute "formname" in the <we:write> tag.

In order to allow a visitor to a Web site to change a webEdition document or object, use the <we:a> tag and set the attribute *edit* to the value *document* or *object*. If the tag is used in a <we:listview> tag, the proper listview entry is used automatically as a reference for it. In order to ensure that only the creator of a webEdition document or object can change them, a field must be set in the webEdition document or object in which the ID of the creator is stored.

If you have installed the Customer Management Module, you can set the name of the aforesaid field as a value of the *userid* attribute in the <we:write> tag. In addition, you can give the name of a field from the Customer Management Module as a value for the attribute *admin*. If a value (except 0) is entered in this field for a customer, this user is allowed likewise to change the webEdition document or object. By using the <we:ifRegisteredUserCanChange> tag, you can determine whether or not a customer is allowed to edit a webEdition document or object. The two attributes *userid* and *admin* have the same meaning as with the <we:write> tag.

If you wish to enter a category, you can use the <we:categorySelect> tag. This is accomplished using an easy drop-down menu with which you are able to choose the desired category. An end tag must be used with this tag, even if you have set no <option> tags.

The input can also be saved in a PHP session. This means that the input will not be lost, even if you refer to another page from time to time. The only conditions are that the form must be sent to a webEdition document that contains a <we:form> tag in which either *document* or *object* has been set as a value of the attribute *edit*, and that the <we:sessionStart> tag appears at the beginning of every template. The data of the session will be deleted automatically by a <we:write> tag.

4.6.5 Including documents (<we:include>)

If there are components on your Web site that appear on all templates/pages (such as a navigation system), you can use webEdition to store them in a file and then to include the file using the <we:include> tag.

Create a template for the document (i.e. a simple link list) that you wish to include in another document. Create the source code as an HTML document using the template you just created. Let's assume that the document you wish to include has the ID 100. Next, write <we:include id="100"/> at the point in the main template where you want the document to be inserted. Any change you might make now to the document will be automatically applied to the documents that use the main template, assuming that the documents are dynamic. If you have static documents, you must perform a rebuild before the change can be applied. This feature allows you to create all kinds of dynamic navigation features—even using DHTML—which can then be expanded by an editor, even if he or she does not know any programming languages.

The left-hand navigation system on all our sample pages was created using the <we:include> tag.

Example:

```
<td class="normal" width="27"></td>
<td colspan="2" class="normal" valign="top">
<we:include id="90"/></td>
<td bgcolor="white" colspan="2" valign="top">
```

Here is the code for the finished file navigation.html (ID=90) that you want to insert:

```
<p><a href="/root/index.php"
style="text-decoration:none">News</a></p>
<p><a href="/root/filmberichte/all.php" style="text-
decoration:none">Filmberichte</a></p>
<p><a href="/root/programm/programm.html" style="text-
decoration:none">Programm</a></p>
```

```
<p><a href="/root/links/links.html" style="text-
decoration:none">Links</a></p
```

Here is the result in /root/index.php:

```
<td class="normal" width="27"></td>
<td colspan="2" class="normal" valign="top"><p><a
href="/root/index.php"
style="text-decoration:none">News</a></p>
<p><a href="/root/filmberichte/all.php" style="text-
decoration:none">Filmberichte</a></p>
<p><a href="/root/programm/programm.html" style="text-
decoration:none">Programm</a></p>
<p><a href="/root/links/links.html" style="text-
decoration:none">Links</a></p></td>
<td bgcolor="white" colspan="2" valign="top">
```

4.6.6 Search feature

The webEdition database architecture makes it easy to create a search feature for all your pages. You do not need to index the entire Web site, as you have to do for most search tools.

Begin by creating a results page where the search results can be displayed. This page is a normal summary page (see Section 4.6.2.2, "Creating a summary of all your pages" on page 142). Let's assume that the ID for the results page is 100. Next, insert the <we:form> tag with the following attribute <we:form id="100"> into your template at the point where you want the search input box to appear. Afterwards, write in the we:tag for the search field itself: <we:search type="textinput" size="50"/>. Now you can insert a normal HTML submit button: <input type="submit" value="suchen">, followed by the <we:form> end tag: </we:form>. For example:

```
<we:form id="100">
  <we:search type="textinput" size="50"/>
  <input type="submit" value="search">
</we:form>
```

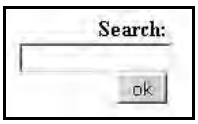
If you enter something in the search field on the finished page and then click the *Search* button (or pressing *RETURN* on your keyboard), the system directs the search query to the results page you created earlier, which will display the search results.

Below is a sample from our sample Web site (/news/news.tmpl):

```
<td colspan="2" align="right">
<we:form id="114" method="get"><span
class="normal"><b>Search:</b><span
class="normal"> </span>
<we:search type="textinput" size="15"/><span
class="normal"> </span><input type="submit" value="OK"><span
class="normal"> </span>
</we:form></td>
```

Figure 175 shows the input box created by this code:

Figure 175 Sample input box



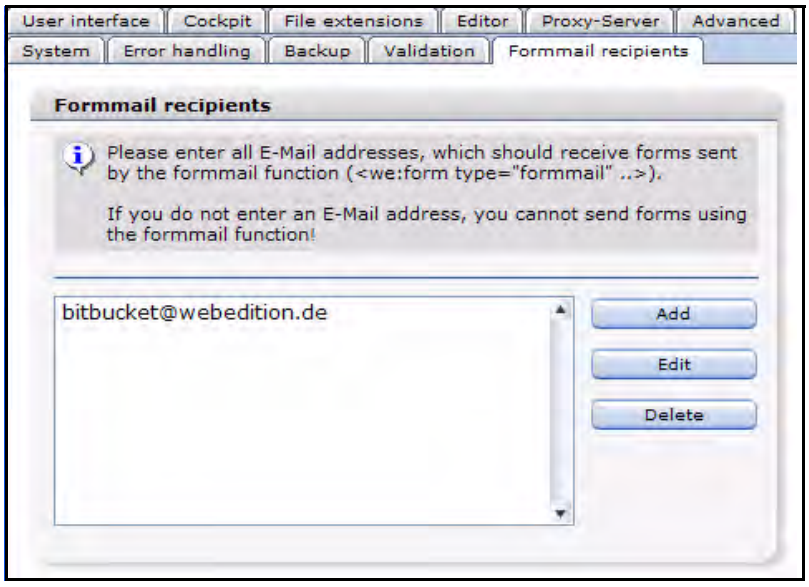
4.6.7 E-Mailing a form

You can use webEdition to create forms that can be sent to an E-mail address. The basic component of this feature is the <we:form> tag. Begin by creating a form using HTML tags. However, instead of inserting the HTML <form> tag, use the <we:form> tag. Insert the type="formmail" attribute in the <we:form> tag. You also must insert the recipient attribute to specify where the E-mail address will be sent. You can also use the *required* attribute to specify which of the fields in the form have to be filled in. For this attribute, separate the names of the fields using commas. The *order* attribute determines in which order the fields will be listed in the E-mail to be sent. Delimit the names of the fields using commas for this attribute. Use the *onerror* and *onsuccess* attributes to provide the IDs of the pages you want to display if there are any erroneous entries (for example, not all the required fields were filled in), or if all the entries are correct. If you want the E-mail to use a specific subject, enter it using the *subject* attribute. If you choose to use the mimetype="text/html" attribute, the mail will be sent in HTML format. Only use this attribute if you know that the recipient of the E-mail has an E-mail program that can display HTML E-mails. If you create a form field with the name "E-mail", this value will be written to the sender address field in the E-mail. webEdition can only write to the sender address field if the address has been entered properly. To ensure that the address is correct before sending the E-mail, webEdition tests the address field and issues a warning if the address is incorrect. For example:

```
<we:form type="formmail" recipient="email@domain.com"
required="email,name,forname" order="name,forname,email"
subject="Form from
my web site" onerror="66" on success="23" mimetype="text/html">
<p>Email: <input type="text" name="email"></p>
<p>Name: <input type="text" name="name"></p>
<p>Forname: <input type="text" name="forname"></p>
<p><input type="submit" value="Send"></p>
```

Note: webEdition has a security feature that prevents the sending of unsolicited E-mail (spamming). By selecting *Options* > *Preferences* > *Formmail recipients*, a view opens in which you can enter the E-mail addresses to which formmails are to be sent. All other addresses will be blocked.

Figure 176 The *Formmail recipients* view



New addresses can be entered by clicking on the add button. To edit addresses, click the appropriate address, then click the *Edit* button. To delete an address, click the appropriate address, then click *Delete*.

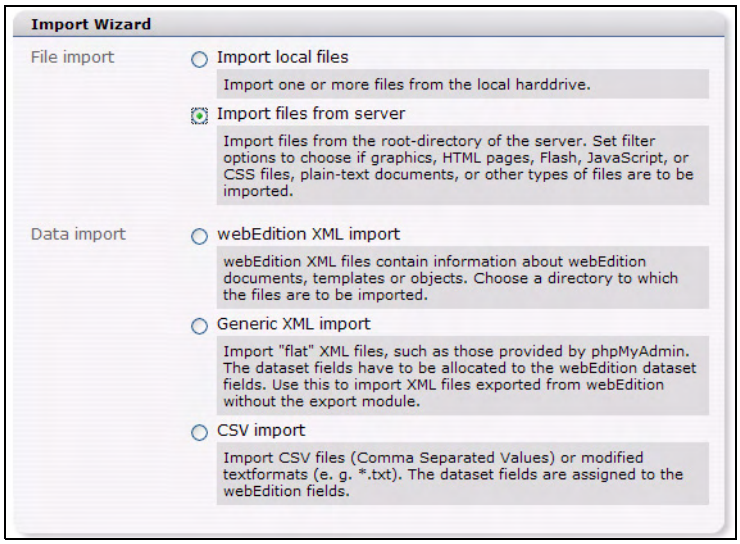
4.6.8 Working with HTML frames

If your Web site uses HTML frames, you can use webEdition to display the content of each individual frame set. You can either directly upload the frame sets to the server yourself (without using webEdition) or you can program them as HTML pages in webEdition. You can use the <we:ifNotWebEdition> tag within the webEdition editing program to eliminate any JavaScripts that load a frame set that belongs to a page or that remove frame sets that do not belong to a page. The code located between the start tag and the end tag will not be executed or displayed when you edit the page in webEdition.

4.7 Importing a Website

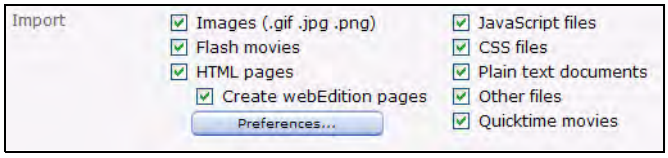
Select *File* > *Import/Export* > *Import...*. The Import Wizard opens (see Figure 177)

Figure 177 Import Wizard



Select *Import files from server* and click *Next*. In the following dialogue, the *Create webEdition pages* checkbox has to be activated.

Figure 178 Checkbox bei Import



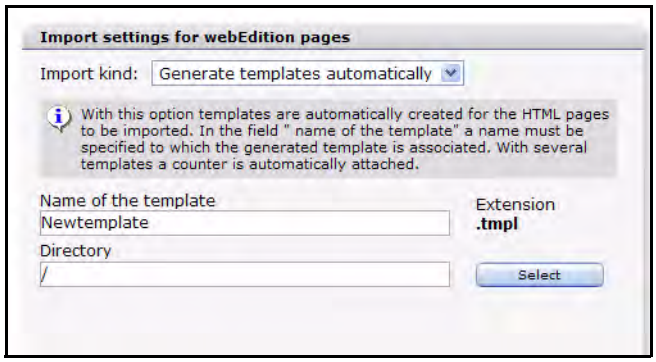
Thus it is possible to import a complete website automatically.

There are two different ways to import. Select the import with the *settings* dialogue, which can be accessed by clicking on the *Preferences...* Button. This button is located underneath the *Create webEdition pages* checkbox (see above).

4.7.1 Generating templates automatically

webEdition automatically generates a template for the HTML page to be imported. All contents between the <body> start tag and end tag are imported in the field *content* and are inserted in a WYSIWYG field in the template (`<we:textarea wysiwyg = "true" name = "content" ...>`).

Figure 179 Automatically generating templates



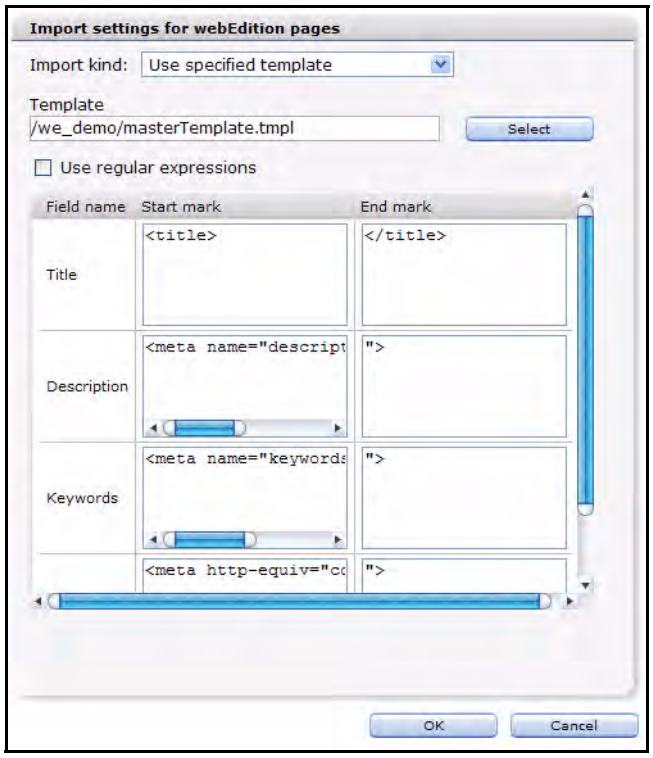
Title, *Keywords*, *Description* and *Charset* are extracted from the HTML document and written in the suitable fields. In the template, the tags are set accordingly (<we:title>, <we:description>, <we:keywords> and <we:charset>). With this import, the target directorys and name can be set which are assigned to the generated template. With several templates, a counter is automatically attached to the name.

4.7.2 Using a dedicated template

webEdition uses a template selected by you and imports sections from the HTML file in the corresponding fields. Which sections should be imported into which fields is determined by a start- and end mark for every field. webEdition then searches the pages to be imported for these selections and imports everything between start and end mark in the suitable field.

For this kind of import, a template has to be selected. Click on *Select* and choose a template. The window then is adjusted according to your selection.

Figure 180 Specified template



After the selection of the template, the fields of the template are displayed. For these fields, a start and end mark can bedetermined for each case. A start and end mark means a unique code fragment directly before and after the field contents to be imported.

Example

All your static web pages are consturcted alike:

```
<html>
<head>
</head>
<body>
<h1>
Überschrift
<h1>
<div id="content">
Hier steht der Content
</div>
<div id="footer">
Hier steht der Footer
</div>
</body>
</html>
```

Your template includes e.g. the following fields:

```
<we:input type="text" name="headline">
<we:textarea name="content">
<we:input type="text" name="footer">
```

Now you enter as start and end mark the following:

- In the field *headline*:
  - Start mark: <h1>
  - End mark: </h1>
- In the field *content*:
  - Start mark:
  - End mark:
- In the field *footer*:
  - Start mark:
  - End mark:

If the Checkbox *Use regular expressions* is activated, you may use regular expressions in the mark fields. You can find further information about regular expressions in the web under *http://en.wikipedia.org/wiki/Regular_expression*

If a date filed has been defined in the template, the date format can be adjusted, too. You can choose from three formats:

- *UNIX Timestamp:* The Unix-Timestamp counts the number of the seconds since the beginning of the Unix epoch (1/1/1970).
- *GMT Timestamp*: General Mean time or Greenwich Mean time (briefly GMT).
- *Individual Format*: Within the formatting the following symbols are valid: Y (four-digit output of the year: 2004), y (two-digit output of the year: 04), m (month with leading zero: 01 to 12), n (month without leading zero: 1 to 12), d (Day of the month; two-digit and leading zero: 01 to 31), j (Day of the month without leading zero: 1 to 31), H (hour in the 24-hour format: 00 to 23), G (hour in the 24-hour format without leading zero: 0 to 23), i (minutes: 00 to59), s (seconds with leading zero: 00 bis 59)

4.8 RSS feeds

The display of contents as a RSS feed becomes more and more popular. With webEdition it is very easy to display external RSS feeds on your page, as well as to display webEdition pages (e.g., current news) as a RSS feed.

4.8.1 Integrating external RSS feed

With the following code an external RSS feed is integrated, analogously to *we_demo*, in webEdition and displayed.

```
<we:xmlfeed name="test"
url="http://www.webedition.de/de/about/news/rss2.xml" refresh="0"
/>
<we:xmlnode feed="test" xpath="/rss/channel">
                <h2><we:xmlnode feed="test"
xpath="title"></we:xmlnode></h2>
                <h3><we:xmlnode feed="test"
xpath="description"></we:xmlnode></h3>
                <ul class="unformattedList">
                  <we:xmlnode xpath="item">
                    <li>
```

```
                    <a href="<we:xmlnode xpath="link"></we:xmlnode
>" target="_blank"><we:xmlnode xpath="title"></we:xmlnode></a>
                      <div class="contentBox">
                        <we:xmlnode feed="test"
xpath="description"></we:xmlnode><br />
                      </div>
                    </li>
                  </we:xmlnode>
                </ul>
              </we:xmlnode>
```

The code is analogous to the one in/we_demo/news/rssFeed.tmpl . In the above example, the RSS-Feed from http://www.webedition.de/de/about/news/rss2.xml is displayed . For further details how to use the we:tags <we:xmlfeed> and <we:xmlnode> please consult the webEdition tag reference.

4.8.2 webEdition pages as RSS feed

Offer your visitors a comfortable way of always stayign informed about the latest news. This example is also based on the we_demo. This offers an easy solution for the news aggregation.

Example

```
<rss version="2.0">
  <channel>
    <title>webEdition Demo</title>
    <link>http://<?php print $_SERVER['SERVER_NAME']; ?></link>
    <description>News der webEdition-Demo-Seite</description>
    <language>de</language>
    <pubDate><?php print date("D, j M Y G:i:s \G\M\T");
?></pubDate>
    <lastBuildDate><?php print date("D, j M Y G:i:s \G\M\T");
?></lastBuildDate>
    <generator>webEdition</generator>
    <managingEditor>demo@webedition.de</managingEditor>
    <webMaster>demo@webedition.de</webMaster>
    <copyright>Copyright © 2004 by living-e AG. Alle Rechte
vorbehalten.</copyright>
    <we:listview type="document" doctype="news - Article">
      <we:repeat>
        <item>
          <title>[<we:field name="newsDate" type="date"
format="d.m.Y" />] <we:field name="headline2" /></title>
          <link>http://<?php print $_SERVER['SERVER_NAME'];
?><we:field name="we_path" /></link>
          <pubDate><?php print date("D, j M Y G:i:s \G\M\T",
substr($GLOBALS['lv']->f('newsDate'), 6, 10)); ?></pubDate>
          <guid>http://<?php print $_SERVER['SERVER_NAME'];
?><we:field name="we_path" /></guid>
        </item>
      </we:repeat>
    </we:listview>
  </channel>
</rss>
```

This code is taken from the template `/we_demo/news/rss2.tmpl`. The specifications comply to the standard RSS 2.0. A summary of all documents of the type “news - Article“ is generated. Based on this template, the XML file `/we_demo/news/rss/rss2.xml` is generated. To offer the feed to the visitors, it is announced in the main template */we_demo/masterTemplate.tmpl*:

```
<!--  Offer subscription to rss-feed -->
```

```
  <link rel="alternate" type="application/rss+xml"
title="webEdition Demo News" href="/we_demo/news/rss/rss2.xml" />
```

The integration in the main template offers the advantage that the feed is advertised on all pages of your Internet presence based on the mastertemplate.

Index

A

accessibility 37, 58, 104, 105
  editing hyperlinks 97
  editing table captions 100
  editing tables 98
  screen readers 97
  validating your code 36
acronym editor for screen readers 97
Active Server pages. See ASP
administrative rights 21
administrators
  knowledge requirements 21, 123
  responsibilities 21
  role of 123
Administrators role
  photo album templates 144
Advanced Search 46
Advanced view
  function described 31
  PHP attributes in we:tags 31
    set to false 32
    set to true 32
archiving pages 144
ASP
  used in templates 54
Attributes area in Flash movie Properties view 86
Attributes area in images Properties view 77
Audience 15
audience profile 15
auto break settings 95
automatic link to next/previous page 141

B

Backup
  create 41
Backup recovery 41
barrier-free web sites 58
  debugging code 36
  features for screen readers 97, 98, 100, 104, 105
  use of XHTML in 37
BIN files, importing 89
Browse Server command 41

C

cascading style sheet. See CSS
Categories
  command 42
  creating 92
  purpose of 42, 92
Categories area in Documents Properties view 55
categories attribute used with list views 143
Change Change password command 42
Change User Name
  command 42
checkboxes
  using in templates 94
CMS
  defined
connecting to a database 32
content images
  defined 20
  using 73
content management system. See CMS
converting image formats
  JPEG 80
  PNG 80
CSV data
  importing 70
Customer service 18

D

Data import functions 40
database
  advantages of 20
    search capability 20
    transparent to user 20

database architecture
  distinguished from file-based systems 19
Database connections
  connect 32
  pconnect 32
Database/Object Module
  interaction with 40
date boxes
  using in templates 94
debug functions
  expert settings preferences 36
debugging accessible web sites 36
deleting
  templates 133
Depth feature
  used for importing directories 68
description of Web page
  displaying with meta tags 55
directories
  copying 64
  creating 64
  displayed in explorer menu 64
  icons 44
  Properties view 64
  thumbnail 34
doctype attribute used with list views 143
Document area in Documents Properties view 55
Document types
  command 41
  creating 90
  default values 89
  deleting 92
  editing 92
  purpose of 41, 89
  used with summary pages 90
documents
  "parking" 61
  deleting 62
  Edit view
    Flash movie 86
    HTML pages 85
    JavaScript 88
  file data in Information view 56
  graphical 73
  in explorer menu 43
  Properties view
    Flash movie 86
    HTML pages 84
    JavaScript 87
    Quicktime movie 87
  saving and publishing 61
  saving without publishing 61
  searchable 55
  unpublishing 61
  updating 64
  viewing unpublished 63
  views 53
    Edit 55
    Information 56
    Preview 57
    Properties 54
Documents Validation view 58
Dynamic navigation entry 113
dynamic pages
  creating 55
  extensions 30, 54
  rebuilding 65

E

Edit view
  in normal mode 48, 55
  JavaScript 88
  with documents 55
  with images 78
  with templates 127
editing
  images 79
  templates 133
Editing hyperlinks for screen readers 97
Editing tables for screen readers 98
Editor Plug-in 30
  license activation 30
  system requirements 30
Editor view
  preferencess 30
editors
  common tasks 53
  knowledge requirements 21
  role compared to administrators 53
  role of 21, 93
  use of seeMode 50
  working with templates 93
EXE files, importing 89
Expanded abbreviations for screen readers 97
Expert settings 36
explorer menu 43
  absent in seeMode 52
  black text in 61
  blue text in 61
  display of documents, objects and templates in 43
  icons in

directory 44
images 44
movie 44
object 45
object class 45
plain text 45
templates 44
webEdition pages 44
ID numbers in 44
interaction with the Database/Object Module 43
red text in 61
webEdition 26
extensions 30
html for static pages 54
extensions php for dynamic pages 54

F

File extensions
HTML extensions 30
webEdition extensions 30
File extensions view
preferences 30
File import functions 40
File menu
described 38
items
Backup 41
Browse Server 41
Delete 39
Import/Export 40
New 39
Open 39
Quickstart 41
Rebuild 40
Save 39
Unpublished items 40
files on server
importing 68
Flash movie
Edit view 86
Properties view 86
Flash movie fields
using in templates 106
Flash movies
fields in templates 106
formatting
thumbnails 81
Formmail recipients
function described 31
forms
emailing 151
fromatting
thumbnails 81
fromatting images 80

G

graphical documents
used in webEdition 73

H

Help menu 42
Update 42
Highlighting 117
HQX files, importing 89
HTML
br tag 95
editing programs 135
frames with seeMode 106
HTML extensions 30
HTML frames
used with webEdition 152
HTML pages
Edit view 85
editing 84
Properties view 84
HTML tags
required in templates 132
HTTP
authentication 33
Http path
in Information view 56
Hyperlink area in images Properties view 78

I

icons
directory 44
images 44
movie 44
object 45
object class 45
plain text 45
templates 44
webEdition pages 44
ID numbers
in explorer menu 44
purpose 44
Identification number. See ID
image fields
using in templates 106
images
cropping 80

descriptive texts for screen readers 105
editing
changing format 80
resizing 79
rotating 79
fields in templates 106
icons 44
importing 75
mouse-over 78
Properties view 77
roll over 78
thumbnail format 81
uploading 73
views
Edit 78
Information 80
import
existing Web pages 20
Import Wizard
using 65
Import/Export functions 40
Importing
files types
BIN 89
EXE 89
Macintosh HQX 89
Microsoft documents 89
other 89
PDF 89
SIT 89
ZIP 89
importing
CSV data 70
data 65
files from server 65, 68
files from the server
recursive directories 68
formatted texts using RTF 100
Generic XML data 69
images 75
local files 65, 66
importing CSV files 65
Import-Wizard 152
include feature
creating a navigation system with 149
including documents in a page 149
Information view
in normal mode 48, 56
with documents 56
with images 80
with templates 127
input boxes
using in templates 94
interaction with User Management Modules 42

J

Java
enable in browser 27
Java Server pages. See JSP
JavaScript
Edit view 88
Properties view 87
JSP
used in templates 54

K

keywords for Web page
displaying with meta tags 55
knowledge requirements
administrators 21
editors 21

L

language preferences
setting 26
languages
installation 43
layout images
defined 20
links to 132
used in templates 132
using 73
link lists
using in templates 102
linklists
creating 139
links
creating 139
editing 103
lists
using in templates 102
local files
importing 66
local host path
in Information view 56

M

Mastertemplate 129
meta tags
Description field 55
Keyword field 55
Title field 55

Meta Tags area in Documents Properties view 55
Microsoft document files, importing 89
Modules menu 42
movie
  icons 44
MySQL
  database architecture 20
  default settings 89
  upload capacity 89

N

naming
  thumbnails 81
navigation control elements 48
Navigation entry
  dynamic 113
navigation entry
  static 111
navigation system
  creating using include feature 149
Navigation tool
  Highlighting 117
  Menus 110
  New items 110
  use of 109
non-graphical documents
  CSS 84
  Flash movies 84
  HTML 84
  JavaScript 84
  Quicktime movies 84
normal mode 48
  views in
    Edit view 48
    Information view 48
    Preview view 49
    Properties view 48
    Scheduler PRO view 49

O

object
  deleting 62
  icons 45
object class
  icons 45
object classes
  deleting 62
objects
  in explorer menu 43
  viewing unpublished 63
offset attribute used to archive pages 144
Options menu
  items
    Categories 42, 92
    Change password 42, 93
    Change User name 42, 93
    Document types 41, 89
    Preferences 42

P

pageLogger 34, 37
  define directory 34, 37
  Systemsettings 34, 37
Passwords
  changing 93
Path area in Documents Properties view 54
Path area in Flash movie Properties view 86
Path area in images Properties view 77
PDF files, importing 89
permissions
  to change a document in the front end 149
persistent connection to a database 32
  explained 32
photo album
  creating templates for 144
  creating templates for multiple albums 146
PHP
  attributes
    default settings 32
plain text
  icons 45
Precautionary messages
  about 16
  Attention boxes 16
  Caution boxes 17
Preferences
  database connections
    types of 32
  System 33
preferences
  Advanced
    Expert settings 36
    HTTP authentication 33
  Advanced view 31
  Editor view 30
  File extensions view 30
    dynamic pages 30
    HTML extensions 30
    static pages 30
    webEdition extensions 30
  Formmail recipients view 31

interaction with the User Management Modules 29
Proxy server view 31
setting 29
tabs and views explained 29
User interface
screen dimensions
screen dimensions
user preferences 29
User interface view 29
Language 29
seeMode start document 29
Preferences window
options in 26
Preview view
in normal mode 49, 57
with documents 57
with templates 128
Properties view
Flash movie 86
for directories 64
in normal mode 48, 54
JavaScript 87
Quicktime movie 87
with documents 54
Categories area 55
Document area 55
Meta Tags area 55
Path area 54
with Flash movies
Attributes area 86
Path area 86
with images 77
Attributes area 77
Hyperlink area 78
Path area 77
with templates 125
Proxy server
function described 31

Q

Quickstart command 41
Quickstart screen for webEdition 27
Quicktime movie
Properties view 87

R

Rebuild command 65
purpose 40
rebuild command
used after including a document 149
reference documentation 15
resizing images 79
rotating images 79
RSS Feed
integrating an external 155
output of webEdition pages as 156
RTF
correcting import errors 102
importing 100
importing formatted texts 100
importing into text boxes 100

S

Save
templates 133
saving webEdition documents
existing pages 61
new pages 61
Scheduler PRO Module
interactions with 49
Scheduler PRO view
in normal mode 49
screen readers 97, 98, 100
script languages
ASP 54
JSP 54
Search
advanced 46
simple 45
Search feature 55
search feature
advantages in webEdition 150
using 150
seeMode
about 49
advantages for editors 50
editing documents in 107
explorer menu 52
interactions with the User Management Module 52
restrictions with HTML frames 106
start document 50
user interface described 51
User Management Module control features 52
using 50, 106
select boxes
using in templates 94, 105
Simple search 45
SIT files, importing 89
sizing

thumbnails 81
start document in seeMode 50
Static navigation entry 111
static pages
creating 55
extensions 30, 54
rebuilding 65
summaries
of specified areas 143
summary of all pages
creating 142
used for navigation 142
summary pages
creating 140
described 140
we:tags for 140
Super-Easy-Edit Mode. See seeMode
system requirements
for Editor Plug-in 30
Systemsettings for pageLogger 34, 37

T

Table captions for screen readers 100
Tag
we
pagelogger 38
Tag Wizard 20, 124
template administration
multiple photo albums 146
photo albums 144
Templates
automatical creation 153
including 131
including of 131
Rebuild 131
using a dedicated 153
templates
about 123
checkboxes in 94
creating 132
creating with HTML editors 135
date boxes in 94
deleting 62, 133
development features
link and linklist 139
Flash movie fields in 106
HTML in 123
icons 44
image fields in 106
in explorer menu 43
input boxes in 94
link lists in 102
lists in 102
modifying 133
reqired HTML tags 132
saving 133
select boxes in 94, 105
text boxes in 95
views
Edit 127
Information 127
Preview 128
Properties 125
we:tags in 123
working with 93
working with fields in 55
text boxes
automatic line break in 95
using in templates 95
thumbnails
directories for 34
formatting 81
naming 81
sizing 81
title of Web page
displaying with meta tags 55
typical tasks of 21
Typographical conventions 17

U

Unpublished objects 40
Unpublished pages 40
Updating webEdition 42
User interface
language preference 29
preferences 29
seeMode start document 29
user interface in seeMode 51
User Management Module
control feature with seeMode 52
uses with seeMode 52
User names
changing 93
interaction with the User Management Modules 93

V

Validation 58
markup validation services 58
services 58
Validation view 36, 37
viewing mode

seeMode 48
viewing modes
normal 48

W

we:a tag 148
we:form 151
we:form tag
changing documents from the front end 146
types 146
we:pagelogger 38
we:tags 151
advantages of 20
as placeholders for variable data 20
attribute values 17
category 124
described 123
description 124
input 93, 123
keywords 124
link 124
link list 94, 124
PHP attributes in 31
search 124
select 94, 124
syntax 123
syntax described 20
Tag Wizard 20, 124
text area 94, 123
title 124
using 123
we:a 148
we:back 141
we:block 102
we:flashmovie 106
we:form 146
we:ifRegisteredUserCanChange tag 149
we:img 106
we:include 132, 149
we:input type="checkbox" 94
we:input type="choice" 94
we:input type="date" 94
we:input type="text" 94
we:link 139
we:linklist 102, 139
we:listview 140
we:navigation 117, 136
we:navigationEntries 118, 137
we:navigationEntry 118, 136
we:navigationField 119, 137
we:navigationWrite 119, 137
we:next 141
we:postlink 139
we:prelink 139
we:repeat 140
we:select 105
we:textarea 95
we:tr 145
we:write 147
winprops attribute 145
wysiwyg attribute 95
we:write tag
used for saving form data 147
we_path field name 142
we_text field name 142
Web pages
adopting 133
adopting portions of 135
description of 55
existing imported into webEdition 20
keywords for 55
previewing 57
title for 55
Web site
adopting an existing 133
adopting portions of 135
webEdition
and MySQL 20
as an HTLM editor 20
checking version 43
content images in 20
described 19
documentation suite 15
explorer menu 26, 43
File menu 38
how it works 19
importing existing Web pages 20
installation 19
language preferences 26
layout images in 20
main screen
returning to 29
main screen described 26
menus 26
Java requirement 27
new features in version 4.0 22
on-line documentation 15
Quickstart screen 27
starting 25
templates described 20
updating 42
user roles 21
we:tags 17, 20

workflow described 22
webEdition documentation
Customer documentation suite 15
on the World Wide Web 15
version and issue 18
webEdition documents
creating 59
creating a new document 39
deleting 39
opening 39
publishing 61
saving 39, 61
unpublished 40
webEdition objects
unpublished 40
webEdition pages
icons 44
viewing unpublished 63
webEdition version
checking 43
webEdition XML import 71
Website
importing 152
winprops attribute 145
word processing applications
used with webEdition 100
WYSIWYG
creating an input screen 95
we:tag attribute for 95

X

XHTML
accesible web sites 37
XML
generic, import of 65
importing generic XML data 69

Z

ZIP files, importing 89

webEdition 4
User Guide

Standard 4.0.1
July 2006
Printed in Germany

© 2006 living-e AG
All rights reserved.

living-e AG
Karlsburgstr. 2
76227 Karlsruhe
Germany

Internet: http://www.living-e.com
E-mail: info@living-e.com