



webEdition User Guide

Database/Object Module



The Database/Object Module

User Guide

Standard 03.00
09 February 2004

© 2004 webEdition Software GmbH
All rights reserved.

Printed in Germany

webEdition Software GmbH 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 webEdition Software GmbH. The purchaser is authorized to make one backup copy of the software. A copy of this software can be requested from webEdition Software GmbH at any time or it can be downloaded at www.webedition.de. Neither webEdition Software GmbH nor suppliers of webEdition Software GmbH 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 webEdition Software GmbH has been informed of the possibility of such damage. In any case of damage, liability of webEdition Software GmbH 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 11

1 Database/Object Module: Introduction 15

- What is the webEdition Database/Object Module? 15
 - Intended users 15
 - What is an object? 15
 - What is an object class? 16
 - What the Database/Object Module can do 16
- Interactions with other webEdition modules 17
- Installation 17
- General information and navigation 17
 - Main menu functions 18
 - Tab functions and the explorer menu 19
 - The Objects tab 19
 - The Classes tab 27
- Understanding the automatic naming policies 30
 - Unique 31
 - Day 31
 - Month (01-12) 31
 - Month (1-12) 31
 - Year (02) 31
 - Year (2002) 31
 - Hours 31
 - ID 31
 - Text 32
- Understanding field types 32
 - Select 32
 - Date 33
 - Textinput 33
 - Textarea 33
 - Image 35
 - Integer 35
 - Float 36
 - Object 36
 - Link 36
 - Href 36
 - Binary 37
 - Combining fields, sorting and deleting fields 37
- Creating/editing vs. displaying objects/classes 38
 - Creating/editing objects/classes: permissions 38
 - Displaying objects in the front end 38

2 Creating classes and objects 41

Creating an object class 41
Creating an object 44
Using workspaces 46
 Assigning a new workspace 46

3 Displaying objects and including objects in templates 51

Displaying objects: an overview 51
Database/Object Module we:tags 53
Creating an object summary 53
Displaying an object in a document using the <we:listview> tag 54
Displaying an object in a document using the <we:object> tag 54

Index 57

List of figures

Figure 1	webEdition main screen	18
Figure 2	Create a new object or class	18
Figure 3	Open an object or class	19
Figure 4	Delete an object or class	19
Figure 5	Objects in the explorer menu	20
Figure 6	Objects Properties view	20
Figure 7	Restricting user access	21
Figure 8	The Edit view for objects	21
Figure 9	The Objects Workspace view	22
Figure 10	The Objects Information view	22
Figure 11	The Preview view	23
Figure 12	Save options	23
Figure 13	Object workspace view	24
Figure 14	The Fields view	26
Figure 15	The Search view	27
Figure 16	The class Properties view	28
Figure 17	The class Edit view	29
Figure 18	The Workspace view	30
Figure 19	Class Information view	30
Figure 20	Automatic naming select box	30
Figure 21	Automatic naming policy: unique	31
Figure 22	Automatic naming policy: Text	32
Figure 23	Field Types	32
Figure 24	Select field type	32
Figure 25	Textinput field type	33
Figure 26	Textarea field type	34
Figure 27	Textarea Attribute	34
Figure 28	Image field type	35
Figure 29	Integer field type	35
Figure 30	Float field type	36
Figure 31	Object field type	36
Figure 32	Link field type	36
Figure 33	Href field type	37
Figure 34	Binary field type	37
Figure 35	Combining fields	38
Figure 36	Creating object classes	41
Figure 37	Restrict access: user name pick list	42
Figure 38	Restricting users to Read only access	42
Figure 39	Object Classes Edit view	43
Figure 40	Assigning a workspace for an object class	44
Figure 41	Creating a new object	45
Figure 42	Assigning a workspace to an object	46
Figure 43	Creating a subclass	46
Figure 44	New documents subdirectory	47
Figure 45	New object	47
Figure 46	Workspace for the new object	47
Figure 47	Add a new workspace	48
Figure 48	Summary of all objects in the events class	48
Figure 49	Display properties for objects in a workspace	49
Figure 50	Viewing an object with the Preview view	49
Figure 51	Summary of objects	49
Figure 52	Selected template	52
Figure 53	Diagram of workspaces and templates	52
Figure 54	Template output	54
Figure 55	Example: hyperlink="true"	54

Figure 56 Displaying an object 55

List of procedures

Procedure 1 Creating a new object class 41

Procedure 2 Creating a new object 44

Procedure 3 Assigning a new workspace, and deleting an old one 46

Procedure 4 Assigning an extra workspace 49

About this document

Purpose of this document

This document treats the webEdition Database/Object Module and how to use it.

You can use this manual to learn:

- what the Database/Object Module is
- how to install the module
- how to use this module to manage objects and object classes
- how to create templates to display objects

Audience

This document is intended for personnel in the following groups:

- Web administrators
- Web editors

Note: The Database/Object Module is recommended for advanced users of webEdition, and for those with a knowledge of Web site architecture and design.

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 Guide*
- *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 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:

<p>ATTENTION</p> <p>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.</p>
--

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:

	<p>CAUTION</p> <p>Database warning</p> <p>If you complete this procedure, your database will be overwritten.</p>
---	--

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 Database/Object Module: Introduction

This introduction is intended to help you familiarize yourself with the webEdition Database/Object Module. This chapter treats what the module does and how to install it. These topics are treated in the following sections:

- Section 1.1, "What is the webEdition Database/Object Module?" on page 15
- Section 1.2, "Interactions with other webEdition modules" on page 17
- Section 1.3, "Installation" on page 17

You can also find information in this chapter about the basic layout and command features for the Database/Object Module:

- Section 1.4, "General information and navigation" on page 17
- Section 1.5, "Understanding the automatic naming policies" on page 30
- Section 1.6, "Understanding field types" on page 32
- Section 1.7, "Creating/editing vs. displaying objects/classes" on page 38

1.1 What is the webEdition Database/Object Module?

The Database/Object Module allows you to organize your web content using object-oriented programming. This module alleviates the problem of maintaining similar items on multiple pages by simplifying the tasks of selecting, distributing, maintaining, modifying and reusing many different data objects (such as addresses or events) that share a similar data structure.

1.1.1 Intended users

The Database/Object Module is intended for advanced users of webEdition, and those with a knowledge of Web site architecture and design. A working knowledge of the webEdition template development processes, and webEdition tags (we:tags) is required in order to use this module.

1.1.2 What is an object?

An object is any kind of data (such as texts, headings, images, numbers, select field), which can be stored in a structured format, connected relationally with other objects, and selected and displayed in templates and/or documents. Objects have no layout information. An object is a unique instance of a data whose structure is defined according to a "template" provided by its class. Each object has its own values for the variables belonging to its class.

1.1.3 What is an object class?

A class is the structural definition of an object. An object class defines what fields are available for a particular object. A class may also be defined as a set of objects that share a common structure and behavior. For example, a customer security database may have the following classes: last name, first name, address, passport photo etc.

1.1.4 What the Database/Object Module can do

The Database/Object module has many possible uses and applications designed to streamline your web content management and maintenance. Below are just a few of the possible uses. With the Database/Object Module, you can manage

- addresses
- events and addresses
- an image database
- a publications database
- contacts
- news
- a web portal

1.1.4.1 Managing addresses

The management of addresses is a typical application for the Database/Object Module. You can define a class called *Addresses* with all the information you wish to have (street, number, postal code, etc.). The editors can enter all the necessary addresses. You can arrange the addresses in different listviews (by postal code, metadata, last names, etc.). You can define two different templates for different areas (such as Internet and intranet), and display the address in each area using the template allocated to this workspace. All information could be displayed on the intranet, while only part of the information would be visible on the Internet.

1.1.4.2 Managing events and addresses

The connecting of events and addresses is an example of the relational linking of objects. Suppose that you have an editorial staff that publishes new events every day. You have defined a link to the *Addresses* object class in the *Events* object class via the *Object* field type. The events keep changing but the addresses of the event organizers remain the same. After entering a new event, editors can select an address already available in the Database/Object Module and allocate it to the new event. If the address changes, everything will still remain up-to-date.

1.1.4.3 Image database

A small image database can also be realized using the Database/Object Module. You can create an object class called "images" in which you can define different field types for a thumbnail of the image, the image itself, or a link to the image (if you prefer not to maintain large images in the database), as well as all keywords and the text descriptions. The object listviews allow you to search this small image database, view different descriptions and thumbnails and download the image itself.

1.1.4.4 Publications database

A publication database operates along the same lines as the image database. You can have publications in different formats (*.doc, *.pdf, etc.). You define an object by

defining all descriptions and metadata for these publications and a download link in the class. The object listview can be used to search and download the publications.

1.1.4.5 Managing contacts

See Section 1.1.4.1, "Managing addresses" on page 16, but with modified fields.

1.1.4.6 Managing news

In this application, you not only use the Database/Object Module, but also the Scheduler Module. You can manage all news items as objects so that you can monitor which news items are included on your page, which have not been published yet, which are online, and which are no longer live.

1.1.4.7 Portal manager

If we assume that you have three portals running at one domain, each one of these portals is a defined workspace in webEdition. The editors from the individual workspaces can create news, information and articles as objects for their workspace. If they have objects they would like to make available to the other workspaces, they can select an additional workspace – that was defined with a fixed template in the class definition – that they can use to make this object available to editors in other workspaces. Editors in other workspaces just have to click on the object to be able to use it for their own portals.

1.2 Interactions with other webEdition modules

The Database/Object Module operates and interacts with all of the webEdition modules you have installed. The User Management PRO, Workflow, and Scheduler PRO modules all have advanced features which enhance the use of the Database/Object Module. If you have not installed these modules, you will not be able to see the advanced feature options.

1.3 Installation

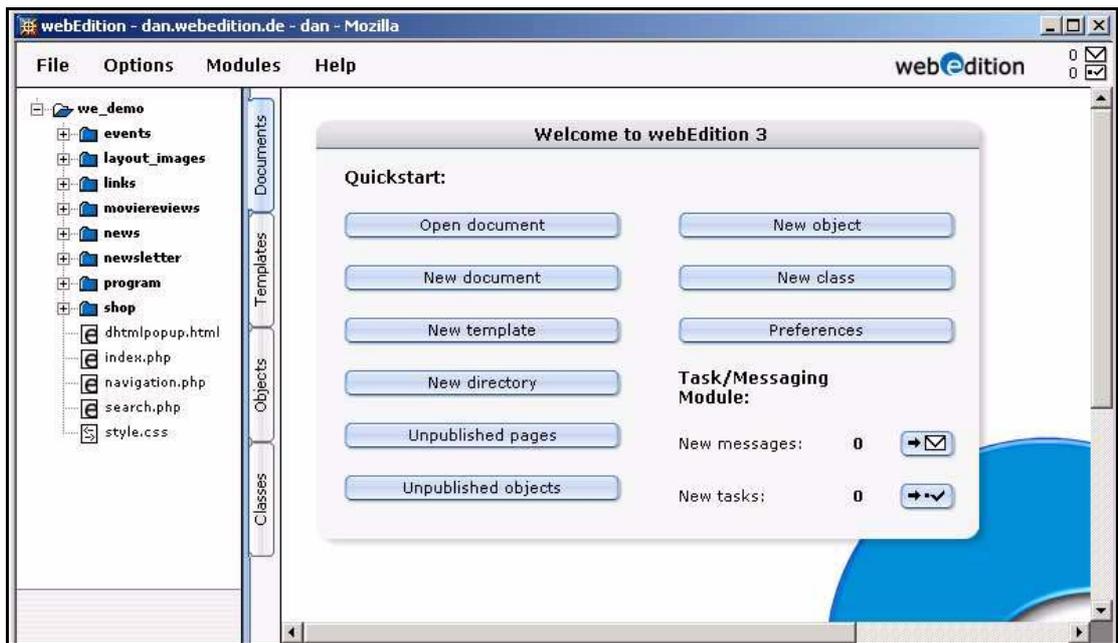
The installation procedure for all modules is described in *The webEdition Installation, Update and Backup Procedures*. A .pdf version of this guide is available at the following URL: <http://www.webedition.de/english/downloads/Documentation.html>

Note: You can also use the webEdition "Live update tool", which is available on the webEdition home page at www.webedition.de.

1.4 General information and navigation

When you install the Database/Object Module, its user functions are integrated into the structure and the *File* menu of the webEdition main screen.

Figure 1 webEdition main screen

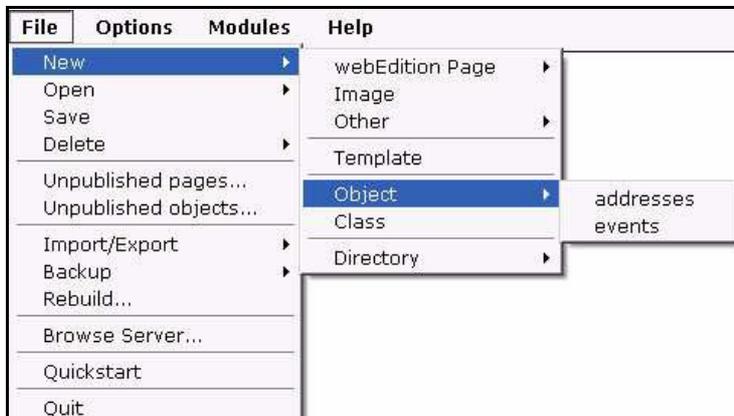


1.4.1 Main menu functions

1.4.1.1 New Object/Class menu

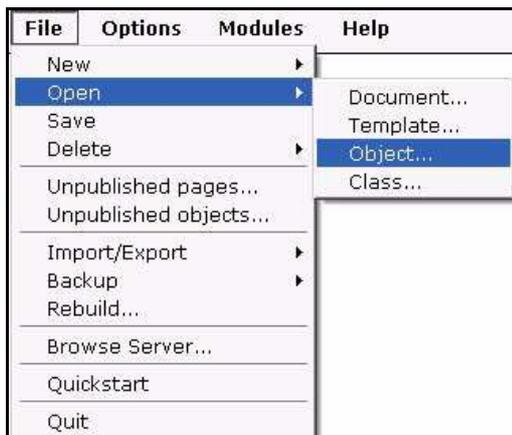
You can create a new object or class of object by selecting the appropriate button from the *Quickstart* screen, or by clicking *File > New > Object* (or *Class*), (see Figure 2).

Figure 2 Create a new object or class



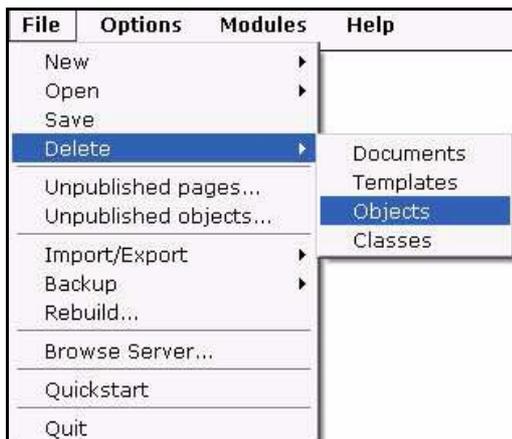
1.4.1.2 Open Object/Class menu

You can open an existing object or class of object by selecting *File > Open > Object* (or *Class*), (see Figure 3).

Figure 3 Open an object or class

1.4.1.3 Delete Object/Class menu

You can delete an existing object or class of object by selecting *File > Delete > Object* (or *Class*), (see Figure 4).

Figure 4 Delete an object or class

1.4.2 Tab functions and the explorer menu

After installing the module, two new tabs, *Objects* and *Classes*, appear along the right-hand side of the explorer menu. The (blue) highlighted tab indicates your present view (see Figure 1, "webEdition main screen" on page 18).

1.4.3 The *Objects* tab

If you click on the *Objects* tab, the tab colour turns to blue, and a list of directories (folders) appears in the objects explorer menu. The "directories" in the objects view represent the object classes. Within each folder are the objects assigned to that object class (see Figure 5).

You can open a folder by clicking the plus "+" symbol. The objects appear in the directory. Objects are identified by an icon with the letter "o". If you click on an object itself, webEdition opens an *Edit* screen, which appears in the right part of your screen.

Figure 5 Objects in the explorer menu

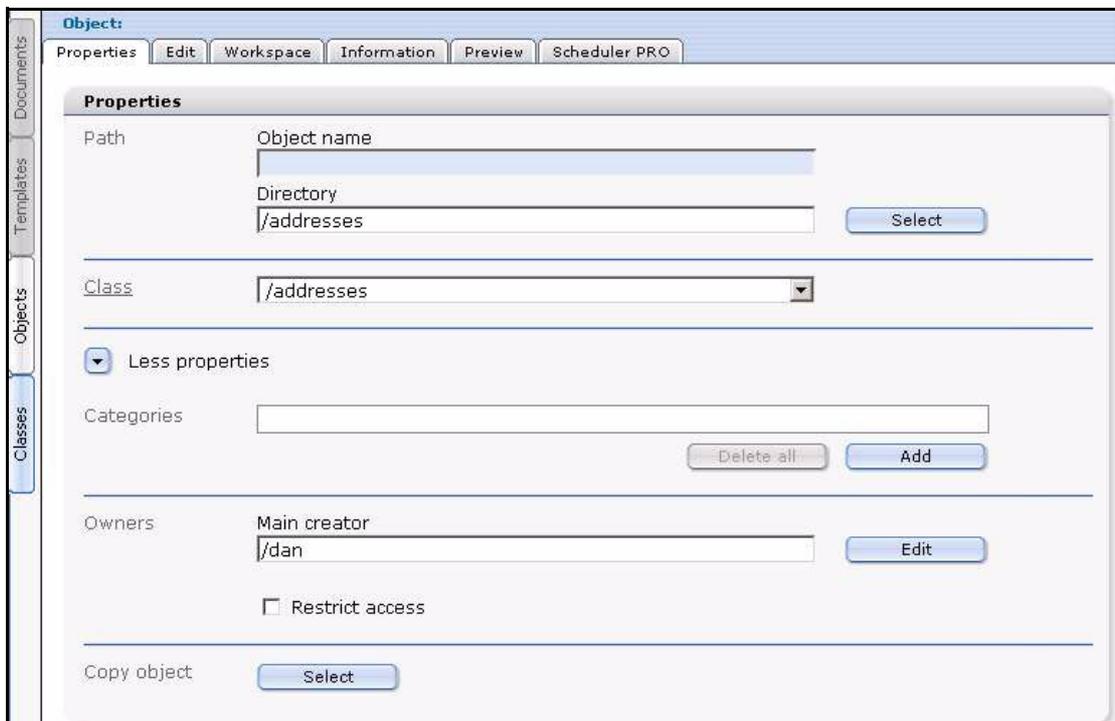


1.4.3.1 Views accessible with the *Objects* tab

The *Objects* tab allows you to view and edit all the characteristics of an object. If you click on the (vertical) *Objects* tab, four horizontally-arranged tabs appear. Each tab opens a view.

1.4.3.1.1 *Objects Properties* view

The *Properties* view has the following areas:

Figure 6 *Objects Properties* view

- *Path*. This area shows the name of the object and its directory location (or address).
- *Class*. This area identifies the class to which the object belongs.
- *Categories*. The categories area allows you to create sub structures so that you can manage your objects.
- *Owners*. The username of the main owner of the object appears in the *Owners* section. (This area is only available if the User Management or User Management PRO Modules have been installed.) If a new object is created, the current user will be the main owner.

- The *Restrict access* checkbox allows you to limit user access to the object. If you activate this checkbox, a dialogue box opens in which you can add those users who are to be allowed to access the object (see Figure 7).

Figure 7 Restricting user access

Restrict access

Grant access for following users.

/axel	<input type="checkbox"/> Read only	
/dan	<input type="checkbox"/> Read only	

Buttons: Delete all, Add

- *Copy object.* The *Copy object* function helps you to create new objects in a simple, quick manner. Because objects share characteristics defined by their class, you can copy an existing object using the *Select* button, then modify and save it under a new name.

1.4.3.1.2 The Objects Edit view *Edit.* The *Edit* view is designed to allow a user to enter data in the fields that have been defined by the selected object class (see Figure 8).

Figure 8 The Edit view for objects

Object: /addresses/SportsHall

Properties Edit Workspace Information Preview Scheduler PRO

Edit

Name1*
Sports Hall of Fame

Name2

Street*
98 Sportstreet

ZIP*
23412

City*
Sportstown

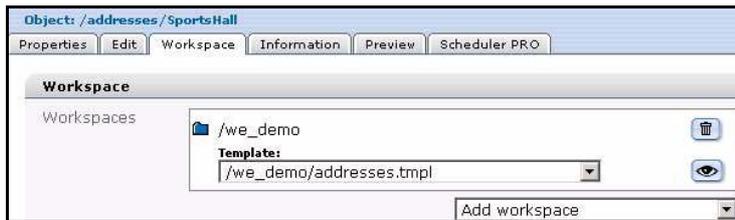
1.4.3.1.3 Objects Workspace view *Workspace.* A workspace is simply a directory to which access rights can be assigned. A workspace contains documents and templates. The term “workspace” has been adopted because it represents the environment in which a user, such as a content editor, performs their editing tasks.

There are two areas to the *Workspace* view:

- *Workspace*, you can see the workspaces specified in the class definition. These are the only areas where the object can be displayed using the corresponding documents.

Only the first workspace defined in the class will be displayed in the *Workspaces* view. You can select any other workspaces defined in the object class in the drop-down menu *Add workspace* (Figure 9).

Figure 9 The Objects Workspace view



- *Extra workspaces*. This area permits you to display an object in an additional workspace.

1.4.3.1.4 Objects Information view *Information*. The *Information* view supplies information about the object identification (ID), file type, the creator, the last change made and the last publication (see Figure 10).

Figure 10 The Objects Information view



1.4.3.1.5 Objects Preview view *Preview*. Use this view to view the appearance of your object (see Figure 11).

Figure 11 The *Preview* view

1.4.3.1.6 Scheduler PRO view *Scheduler PRO*. This tab is available only if you have installed the Scheduler PRO Module. For details, see the *Scheduler PRO Module User Guide*.

1.4.3.2 Save options

If you wish to save a new or modified object, you have the following options (see Figure 12, "Save options" on page 23):

- *Save*. Select the save button to store an object in the system without publishing it. An object that has been saved but not published appears in red font in the explorer menu.
 - *Publish after saving*. Select this checkbox before saving if you want to save the object in the system and publish it immediately. It will be immediately visible in the corresponding listview.
 - *New object after saving*. Select this checkbox before saving if you want a new, empty object from the same class to appear after the save process is complete. This function facilitates the creation of many objects of the same type.
- *In Workflow*. Select this button to place an object into a workflow.

Note: This option is available only if you have installed the Workflow Module. For further details, see the webEdition documentation page at <http://www.webedition.de/english/downloads/Documentation.html>
- *Unpublish (Park)*. use the *unpublish* function to undo the publication of an object. By doing so you can return an object that has been saved and published to the "saved, but not yet published" status.

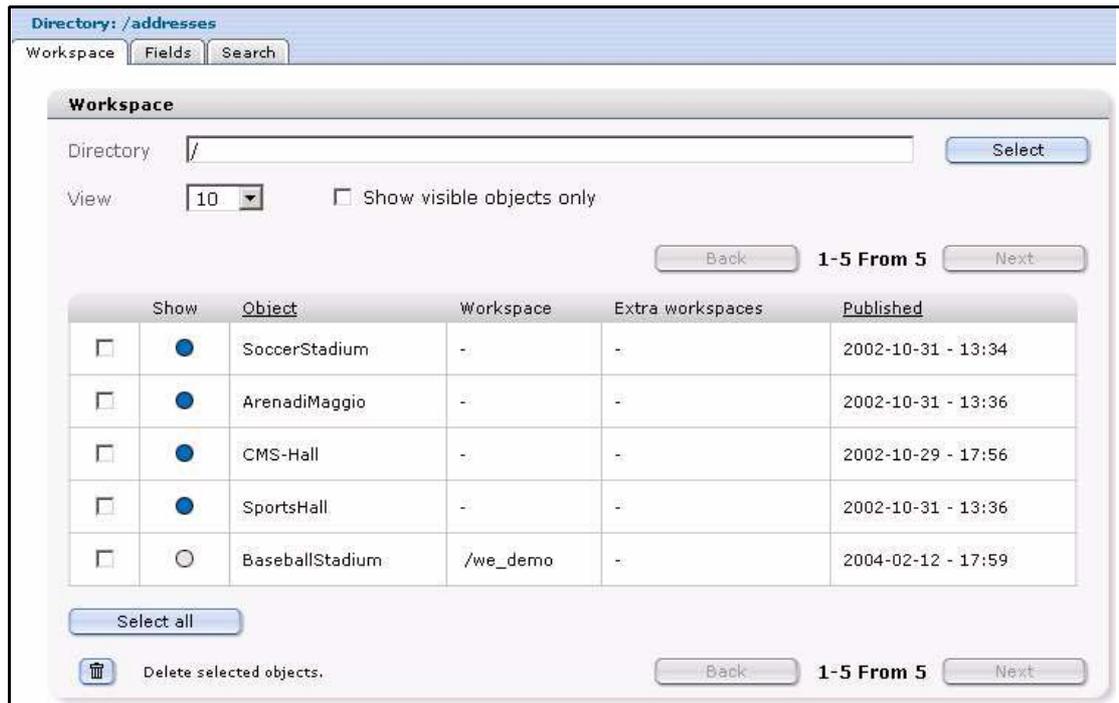
Figure 12 Save options

1.4.3.3 Object class summary

If you click on a folder in the explorer menu (while still in the *Objects* view), you can access summary information about the object class. Three tabs appear as shown in Figure 13, "Object workspace view" on page 24:

- Workspace
- Field
- Search

Figure 13 Object workspace view



1.4.3.3.1 The *Workspace* view The *Workspace* view shows a summary of all objects contained in a class together with all of the assigned workspaces and additional workspaces.

The top portion of the *Workspace* view shows the following items:

- *Directory (field)*. This selector field allows you to select the directory whose workspace you wish to view.
- *View*. This pick list allows you to change the number of objects to display on your screen.
- *Show visible objects only*. Select this checkbox if you wish only to view objects that are visible.

The workspace summary table has the following information and functions:

- *Directory (checkbox)*. The square checkboxes in the first column allow you to select different workspaces. A directory checkbox is used only to determine what objects can be displayed in a given workspace.
- *Show*. The circles in the second column of the table indicate which objects can be displayed in the selected workspace and which cannot.
 - If the circle is gray, the object cannot be displayed.
 - If the circle is blue, the object can be displayed.
- *Object*. This column lists the objects in the selected class. If you click on the word *Object* in the table heading, the list will automatically be sorted alphabetically by object name.
- *Workspace*. This column shows the workspace allocated to an object.
- *Extra workspaces*. This column shows the additional workspaces allocated to an object.
- *Published*. This column identifies the publication status of an object (published, parked or saved) and the date and time of the last operation performed on the object. If you click the word *Published* in the table heading, the list will be sorted by publication date. If an object has not been published or parked, nothing will be entered in the column.

Beneath the table are the following buttons/functions:

- *Select all*. This button allows you to select all directories in the table.
- *Delete selected objects*. The trash can icon allows you to delete the objects you have selected from the table.
- *Search*. You can perform a full-text search by entering a word in the search field. You can also use the adjacent select box to filter your search according to field types.

1.4.3.3.2 The *Fields* view The *Fields* view looks and operates in the same manner as the *Workspace* view, except that it displays all the fields of the object instead of the workspaces (see Figure 14). (Scroll to the right, using the scroll bar at the bottom of the page, in order to see all fields). If you click on the field name in the table heading, the list will be sorted alphabetically according to the entries of that field.

Figure 14 The *Fields* view

Directory: /addresses

Workspace Fields Search

Fields

Directory: /

View: 10 Show visible objects only

1-5 From 5

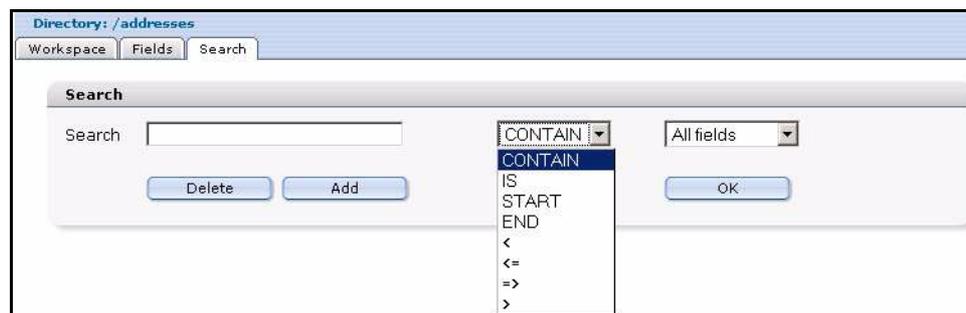
Show	Object	Name1	Name2	Street	ZIP	City	Phone	Email
<input type="checkbox"/>	<input checked="" type="radio"/> SoccerStadium	Soccer Stadium		65 Sportstreet	67868	Sportcity		
<input type="checkbox"/>	<input checked="" type="radio"/> ArenadiMaggio	Arena di Maggio		Via Arena 12	45464	Magicten	009876 / 4389334	info@arenadimaggio.c ...
<input type="checkbox"/>	<input checked="" type="radio"/> CMS-Hall	CMS-Hall		CMS-Street 111	12345	CMS-City	01212 / 38934	cms@cms.cms
<input type="checkbox"/>	<input checked="" type="radio"/> SportsHall	Hall of sports		98 Sportstreet	23412	Sportstown		
<input type="checkbox"/>	<input type="radio"/> BaseballStadium	Baseball Stadium	Balldome	100 Baseball Way	67000	Sportcity	555-5555	baseball@baseballsta ...

1-5 From 5

1.4.3.3.3 The Search view In this view, you can search individual fields using specific criteria (see Figure 15, "The Search view" on page 27).

- *Search.* Enter a keywords into the *Search* field. Next to it is a drop-down menu in which you can select the condition you wish to attach to the keyword.
- Search operator pick list. Use this list to select one of the following search operators:
 - *CONTAIN.* The keyword matches a freely definable combination of characters within the term being searched. Example: "s_t" – returns "news_today" and all other terms that contain this combination of characters.
 - *IS.* 100% match with the keyword, no variation allowed. Example: Keyword: "news_today" – returns: "news_today"
 - *START.* The keyword matches the beginning of the term being searched. Example: Keyword: "news" – returns "news_today" and all other terms that begin with "news".
 - *END.* The keyword matches the end of the term being searched. Example: Keyword: "today" – returns "news_today" and all other terms that end with "today".
 - For numbers:
 - < is less than.
 - > is greater than
 - <= is less than or equal to
 - >= is greater than or equal to
- *Select field.* In the last pick list on the right side, you can select which field should be searched.
- *Delete.* If you want to start a new search, click the *Delete* button to reset the search function.
- *Add.* Click the *add* button if you want to search additional fields simultaneously.

Figure 15 The Search view



1.4.4 The Classes tab

If you click on the *Classes* tab, the tab colour turns to blue, and a list of files appears in the classes explorer menu. The entries in the class view represent the object classes. A class is identified by an icon with the letter "c".

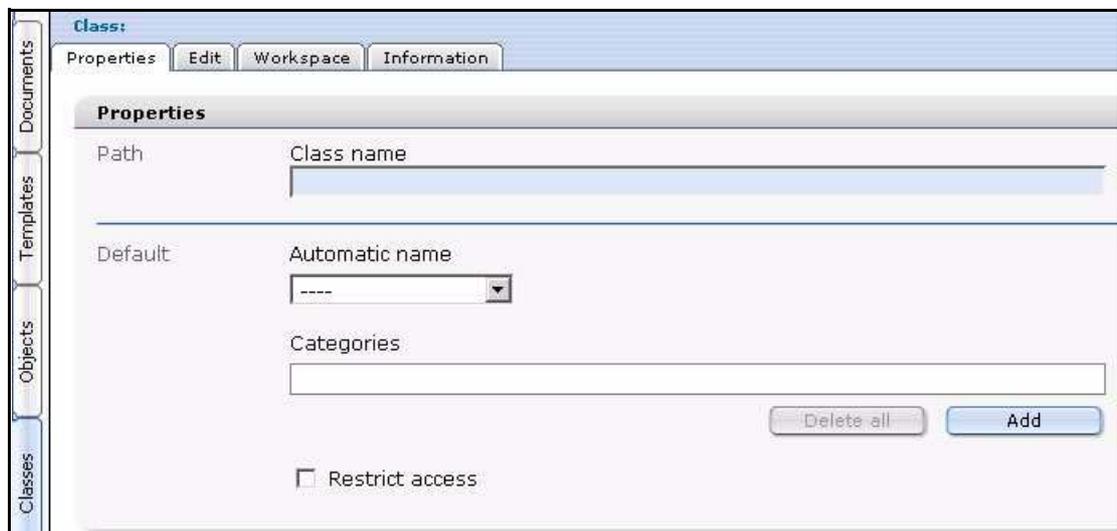
1.4.4.1 Views accessible with the **Classes** tab

You can access the Classes views by clicking on a class icon in the explorer menu. The edit mode has four views:

1.4.4.1.1 Classes Properties view *Properties*. This view has two areas, *Path* and *Default* (see Figure 15).

- *Path*. This area shows the name of the class.
- *Default*. This area has the following fields:
 - *Automatic name*. If you wish to use the automatic naming feature, select an item from the *Automatic name* drop-down menu. Each type is described in detail in Section 1.5, "Understanding the automatic naming policies" on page 30.
 - *Categories*. See page 20
 - *Restrict access*. See page 21

Figure 16 The class *Properties* view



1.4.4.1.2 Classes Edit view *Edit*. The actual structure of the object class is created in the *Edit* view (see Figure 17). The structure of an object class consists of different types of data fields. These fields do not contain any layout information.

Figure 17 The class *Edit* view

Click the plus button to create the first field of the new object class.

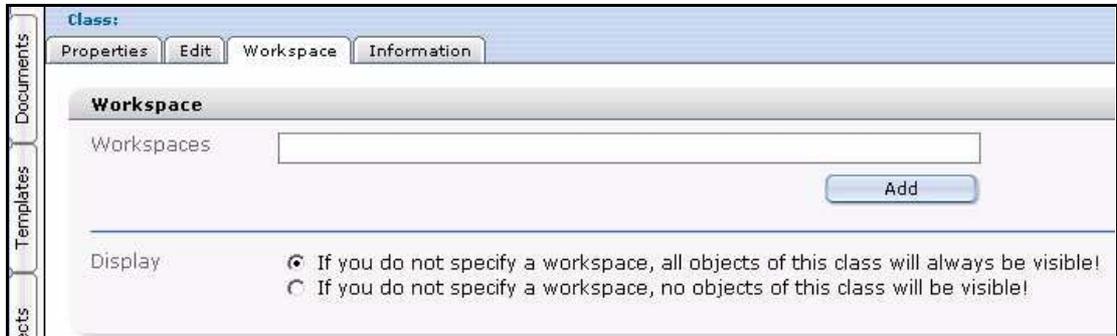
- *Name*. Use this area to specify the name of the field. The following characters are permitted: all letters from a to z and A to Z, numbers, underscore (`_`), minus (`-`) and dot (`.`). No other characters are allowed. Do not use accented letters. You will always be given a default name (`input0`).
- *Type*. Use this area to specify the field type using this drop-down menu. The available field types are described in Section 1.6, "Understanding field types" on page 32.
- *Length*. Enter a value in this text box that represents the length of the field in the database. For example, if you set the value to 20 for a text input field type, the content can only have 20 characters. It is useful for saving memory.
- *Default*. Enter the default value for the corresponding field. When creating a new object, the value that you enter will be filled automatically in the object field.
- *Title*.
- *Description*.
- *Required field*.
- *Users*. Select which users are allowed to access the object by clicking the *Add* button.

1.4.4.1.3 Classes *Workspace* view *Workspace*. In the *Workspace* view, you can pre-set the environment(s) in which the objects based on this object class are visible (see Figure 18, "The *Workspace* view" on page 30). Because the object is stored without any layout information, you must also allocate a template with which to view of the object.

You can also use the radio buttons in the *Display* area to determine whether or not you want to display the object class in the selected workspace.

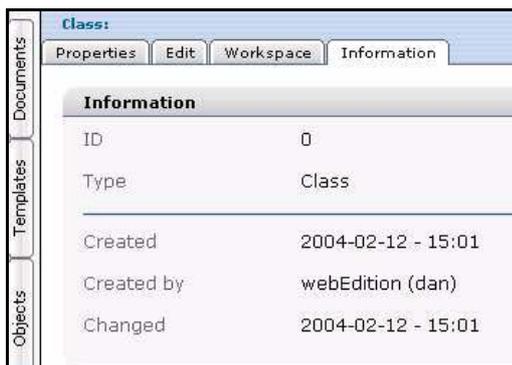
By clicking the add button, you will open the file manager in which you can select a directory (workspace). If you do not define a workspace, the object can be displayed anywhere. The object will always be displayed using a document (listview, detailed view).

Figure 18 The *Workspace* view



1.4.4.1.4 Classes Information view *Information.* The Classes *Information* view supplies read-only information about the file ID and type, the creation date, creator, and the last changes made to the class (see Figure 19).

Figure 19 Class *Information* view



1.5 Understanding the automatic naming policies

The following subsections describe in detail the automatic naming policies. The *Automatic name* options pertain to object classes. The controls are found in the *Default* area of the *Properties* view.

Figure 20 Automatic naming select box



If you fill in the *Automatic name* field, all objects created later with this class will be named uniformly and in order without requiring you to enter a new name each time.

1.5.1 *Unique*

The *unique* option allows you to instruct webEdition to identify an object with a randomly selected and unique alpha-numeric name.

After clicking *unique*, a number field appears beside the *Automatic name* drop-down menu (see Figure 21, "Automatic naming policy: unique" on page 31). This number indicates the number of characters the unique identifier will have. The default value is 16 characters.

Figure 21 *Automatic naming policy: unique*

The image shows a user interface element titled "Automatic name". It consists of a dropdown menu currently displaying "Unique" and a text input field to its right containing the number "16". Below the dropdown menu, there is a partially visible option labeled "----".

1.5.2 *Day*

The *Day* option allows you to specify the portion of the name that will have the current date as its identification. "01" will be displayed if the current date is 09/01/2003.

1.5.3 *Month (01-12)*

The *Month (01-12)* option allows you to specify the portion of the name that will have the date of the current month as its identification. "09" will be displayed if the current date is 09/01/2002.

1.5.4 *Month (1-12)*

As above in *Month (01-12)*, except without the "0" for the months from January ("1" instead of "01") to September ("9" instead of "09"). "9" will be displayed if the current date is 09/01/2003.

1.5.5 *Year (02)*

Clicking this option allows you to specify the portion of the name that will have the date of the current year as its identification. "02" will be displayed if the current date is 09/01/2002.

1.5.6 *Year (2002)*

As above in *Year (02)*, except the full year will be displayed (2002). "2002" will be displayed if the current date is 09/01/2002.

1.5.7 *Hours*

The *Hours* option allows you to specify the portion of the name that will have the current hour as its identification. If, for example, the system clock is currently 11:23, "11" will be displayed. If you set the North American time format (am/pm), this option will still display the hours 0-23.

1.5.8 *ID*

Click the *ID* option and you can specify the portion of the name that will have the ID of the corresponding object as its identification. Like all documents, all objects have a

unique ID. This is also a unique identifier. If the ID of the newly created object is "566", "566" will be displayed.

1.5.9 Text

The *Text* option allows you to specify the portion of the name that will have a text of your choice as its identification (see Figure 22).

This identifying text can contain upper and lower-case letters and numbers. The following characters are permitted: all letters from a to z and A to Z, numbers, underscore (_), minus (-) and dot (.). No other characters are allowed. Do not use accented letters such as ä or ö.

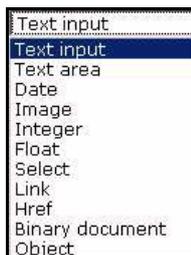
Figure 22 Automatic naming policy: Text



1.6 Understanding field types

The following subsections describe in detail the field types that are available from the *Classes > Edit* view.

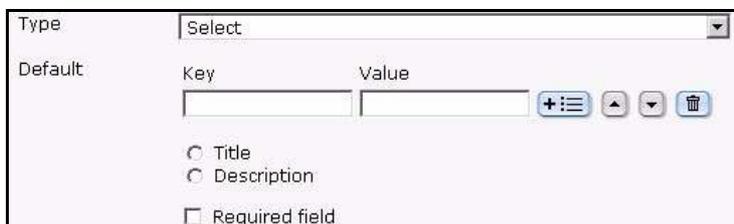
Figure 23 Field Types



1.6.1 Select

The *Select* field type allows you to create a select box or pick-list. Editors can choose an entry from the select field when creating a new object (see Figure 24).

Figure 24 Select field type



The *Default* settings are as follows:

- *Key*. Should begin with 0. Permits future changes to the individual select items under *Value*.
- *Value*. Name of the select item
- *Title*. Fetches the title for search function

- *Description*. Fetches the description for search function
- *Required field*. This field must be filled in; if it is not filled in you will not be able to save the object.

To define a new entry in the select field, click the plus “+” icon next to the *Value* field. To delete an existing entry, click the trash can icon.

The *add* button underneath the *user* field allows you to set the permissions for individual users and/or groups to edit this field.

The two option buttons *Title* and *Description* can only be clicked once per object class. They allow you to define the title and description displayed in a listview (title and description) for each object class. These fields will then be the only ones displayed in the listview.

1.6.2 Date

The *Date* field type allows you to generate a date field. The default setting is always the current date and time. The editor can change this field.

All other aspects are like *Select*.

1.6.3 Textinput

The *Textinput* type permits you to define one-line text entry fields (see Figure 25).

Figure 25 *Textinput* field type

Type	Text input
Length	255
Default	
	<input type="radio"/> Title <input type="radio"/> Description <input type="checkbox"/> Required field

- *Length*. The maximum number of characters that can be entered.
- *Default*. Entry of a default text.

All other aspects are like *Select*.

1.6.4 Textarea

The *Textarea* field type allows you to generate a textarea (that is a textbox with several rows).

Figure 26 Textarea field type

- *Name*: Give the settings a name.
- *Type*: Select *Textarea*.
- *Default setting- Attributes...* When you click on the button *Attribute...*, a new dialogue window for the attribute settings opens (see Figure 27).

Figure 27 Textarea Attribute

- *wysiwyg*: on/off (de)activates the WYSIWYG mode.
- *forbidhtml*: on/off allows or permits to code HTML in the textarea.
- *forbidphp*: on/off allows or permits to code PHP in the textarea..
- *xml*: If *xml* is true, XML conform source code is generated.
- *removefirstparagraph*: If *removefirstparagraph* is true, the first paragraph is removed.
- *inlineedit*: If *inlineedit* is true, the WYSIWYG Editor is deactivated in the normal edit mode. Instead of the WYSIWYG editor, the text is displayed with a Edit button. If you click on that, the WYSIWYG editor appears in a separate

window. Loading a page with a lot of WYSIWYG textareas is much faster this way.

- *showmenus*: on/off (de)activates the drop-down menus (dhtmledit has to be “on” for that function).
- *classes*: Enter the classes, for which the settings apply.
- *commands*: Choose the commands in the select box, for which the settings apply. Than the selected commands are displayed automatically in the textbox underneath. You can also write the commands directly in the textbox.
- *Default setting- Title/Description*: Here you can input a title or a description of a textarea.
- *Default setting- Obligational field*: Choose if text has to be input in the textarea (*Obligational field* activated) or not (*Obligational field* not activated).
- *User*: Define with the button *Add*, which users may input or edit text in the textarea.

1.6.5 Image

This field type permits you to define image fields (see Figure 28).

Figure 28 Image field type



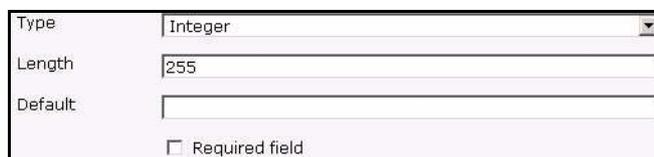
Default setting: If you wish, you can define a default image.

All other aspects are like *Select*.

1.6.6 Integer

This field type allows you to define a number field for integer values (whole numbers) as seen in Figure 29.

Figure 29 Integer field type



- *Length*. The maximum length of number
- *Default* setting. If you wish, you can define a default value.

All other aspects are like *Select*.

1.6.7 Float

This field type allows you to define a number field for float values (decimal numbers) as seen in Figure 30.

Figure 30 Float field type

- *Default* setting. If you wish, you can define a default value.

1.6.8 Object

This field type allows you to link another object to the newly created relational one. The editors will then be able to select objects from an object class whenever they create new objects. For example, if you create objects of the "Contact person" class, you can connect these objects with objects of the "Addresses" class so that contact people are always displayed alongside their addresses (see Figure 31).

Figure 31 Object field type

- *Default* setting. Select the desired class.

All other aspects are like *Select*.

1.6.9 Link

The *Link* field type permits you to define an external link (URL) on another Web page, an internal link or a document to be downloaded (see Figure 32). This field has the same functionality as the <we:link> tag. The output is a link.

Note: External links have to be entered with "http://...".

Figure 32 Link field type

All other aspects are like *Select*.

1.6.10 Href

This field type permits you to define an external link (URL) on another web page, an internal link or a document to be downloaded (see Figure 33, "Href field type" on page 37). This field has the same functionality as the <we:href> tag. The *Select* button for

ext.href launches the "Browse server" function so that you can easily link files that are stored outside of the webEdition database on the server. The output is a link.

Note: External links have to be entered with "http://...".

Figure 33 Href field type

Default: Desired internal or external URL

All other aspects are like *Select*.

1.6.11 Binary

The *Binary* field type allows you to insert a binary file to be downloaded (see Figure 34). This file first has to be imported using *File > New > Others > Other files* in webEdition. The output is a link.

Figure 34 Binary field type

Default: Desired file.

All other aspects are like *Select*.

1.6.12 Combining fields, sorting and deleting fields

You can combine, sort, or delete fields using the add button (plus icon), as seen in Figure 35.

Figure 35 Combining fields

The screenshot shows a web form for creating a field. The form is titled 'Name' and contains the following fields and controls:

- Name:** A text input field containing 'NewField'. To its right are icons for adding, deleting, and moving fields.
- Type:** A dropdown menu showing 'Text input'.
- Length:** A text input field containing '255'.
- Default:** An empty text input field.
- Options:** Two radio buttons labeled 'Title' and 'Description', and a checkbox labeled 'Required field'.
- Users:** A text input field containing 'Everybody' with a user icon on the left.
- Buttons:** 'Delete all' and 'Add' buttons at the bottom right.
- Bottom Left:** A plus sign icon in a small box.

Click the plus button to insert an additional field in front of a field. Click the trash can icon to delete the field.

Click the plus button on the bottom left of the page to add a field.

1.7 Creating/editing vs. displaying objects/classes

In the chapters that follow we shall differentiate between creating and editing objects/classes (see Chapter 2) and displaying them in the front end system (see Chapter 3).

1.7.1 Creating/editing objects/classes: permissions

In general, creating new classes or objects is a privilege that depends on the permissions granted to each webEdition user by the administrator. As such, it can be assigned and withdrawn. You can define in User Management PRO whether someone can create a new class or a new object based on an existing class.

When creating a new class, the administrator can define who can create objects within this new class in the *Properties* view of the object classes. If the administrator does not restrict access, everyone who principally has this privilege will be able to create new objects based on this class. However, an administrator can also restrict access so that only a selection of users will be entitled to create new objects based on this class.

ATTENTION

The permissions assignment function in the User Management PRO Module is the only way to manage the privilege to create object classes. If you have not installed this module, all users will be able to create classes and objects.

1.7.2 Displaying objects in the front end

The display of objects in the front end as a list or in the detailed display, is always done by means of templates and documents. If you want to display stored objects, you must create a document based on a template that contains the corresponding listview or fields for a detailed display.

Unlike creating or changing objects, displaying the objects in the front end system (via listview or the detailed display) is not organized by users, but by workspaces. A workspace is a directory to which access rights can be assigned when you set up your object classes.

For example, if you have a workspace called "PR" and have allocated the "Press releases" object class exclusively to this workspace, objects of the "Press releases" class can only be displayed by a document stored in this area with the corresponding listview. If the same document with the same listview is stored in another directory (workspace), the listview will not display any objects.

2 Creating classes and objects

The following procedures will guide you through the process of creating an object class, and an object. In addition, at the end of the chapter you will find exercises to help you use workspaces.

2.1 Creating an object class

Before you can create any objects, you must create an object class. A class is the structural definition of an object. An object class defines what fields are available for a particular object.

Procedure 1 Creating a new object class

In the webEdition main menu

- 1 In the main menu, click *File > New > Class*.
The Classes Properties view opens (see Figure 36).

Figure 36 Creating object classes

- 2 In the *Path* area, enter the name that you wish to give to the new object class in the *Class name* field.
- 3 If you wish to use the automatic naming feature, select an item from the *Automatic name* drop-down menu. Each type is described in Section 1.4.4, "The Classes tab" on page 27.
If you fill in this field, all objects created later with this class will be named uniformly and in order without requiring you to enter a new name each time.
- 4 If you wish to assign a category to your object class, click the *Add* button.
A dialog box opens in which you can choose a category for the object class.

- 5 If you wish to limit user access to the object class, activate the *Restrict access* checkbox.

Note: The *Restrict access* feature is only available if the User Management or User Management PRO Modules has been installed.

Figure 37 Restrict access: user name pick list

- a Select which users are allowed to access the object by clicking the *Add* button.
A dialog box opens in which you can choose the users who will have access rights to the object class.
- b Select the users that are to have access and click on the *OK* button.
- c If you wish to allow read-only access to the selected users, activate the *Read only* checkbox adjacent to the user name (see Figure 38).

Figure 38 Restricting users to *Read only* access

- d Save the user setting by selecting the *Save* button at the bottom of the page.
An information box opens and webEdition informs you that your changes have been saved.
 - e Select the *OK* button on the information box to continue.
- 6 Click on the *Edit* tab.
The Edit view opens. The first time you create an object class, webEdition displays an empty data field icon.
- 7 Click on the plus "+" icon to open the *Edit* view for the first field of the object class.
webEdition opens the Edit view (see Figure 39).

Figure 39 Object Classes *Edit* view

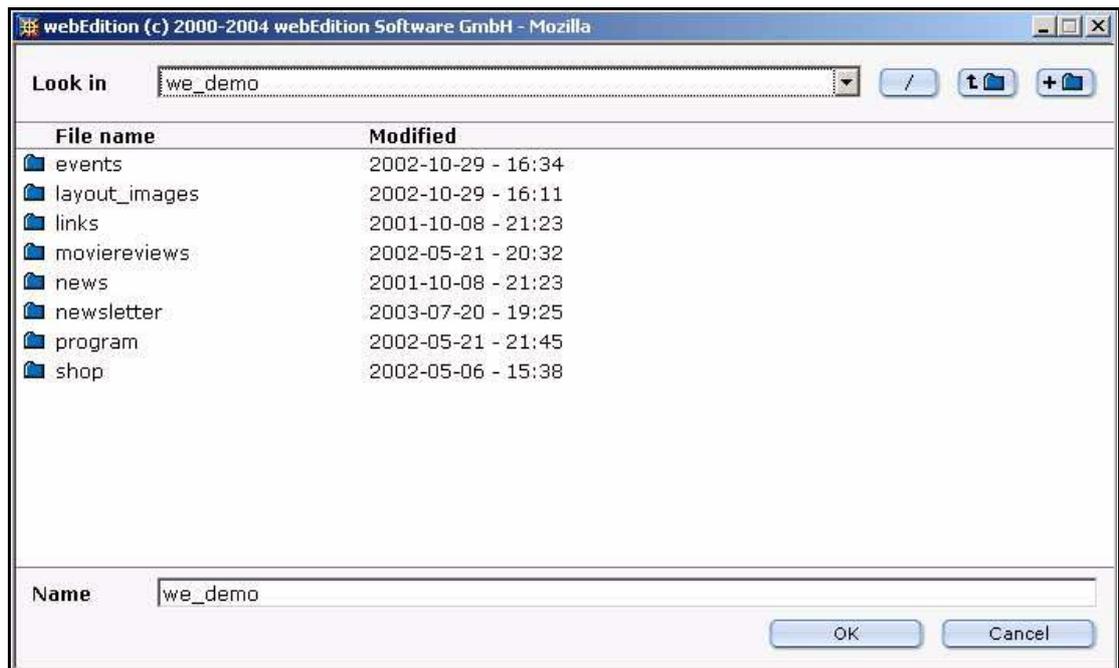
- 8 Specify the name of the field in the *Name* text box.

Note: The following characters are permitted: All letters from a to z and A to Z, numbers, underscore (_), minus (-) and period (.). Do not use accented letters.

- 9 Choose the field type using the *Type* drop-down menu. Each field type has different default settings that you must define. For details about each field, see Section 1.6, "Understanding field types" on page 32.
- 10 You can add, sort or delete fields by doing the following sub-steps:
- To add a new field in front of your current field, click the plus button.
 - To add a second field after your current field, click the plus button on the bottom left of the page.
 - To move a field within the field list, click the up or down arrow.
 - To delete a field, click the trash can symbol to delete this field.
- 11 Click the *Workspace* tab to open the *Workspace* view.
- 12 Assign the workspace(s) for which the objects in this object class are to be visible by clicking the *Add* button.

The file manager opens in which you can select a directory (workspace), as seen in Figure 40.

Figure 40 Assigning a workspace for an object class



Note: If you do not specify a workspace, you can choose from the buttons in the *Display* area whether or not to display an object. The object will always be displayed using a document (listview, detailed view).

- 13 After choosing a directory, you must also allocate a template with which the object (which is stored without any layout information) can be displayed. Select a template from the *Template: pick list* that appears under the directory name.
- 14 Click the *Save* button.
- 15 You have completed this procedure.

2.2 Creating an object

After creating an object class, you can create an object. An object is a unique instance of a data whose structure is defined according to its class. Each object has its own values for the variables belonging to its class.

Procedure 2 Creating a new object

In the webEdition main menu

- 1 In the main menu, click in the menu *File > New > Object > "name of a selected object class"*.

Note: You cannot create an object without first defining an object class. If you have not defined any classes, you will not see any object classes listed in the menu after *Object*. In this case you must return to Procedure 1, "Creating a new object class" on page 41.

The Properties view opens.

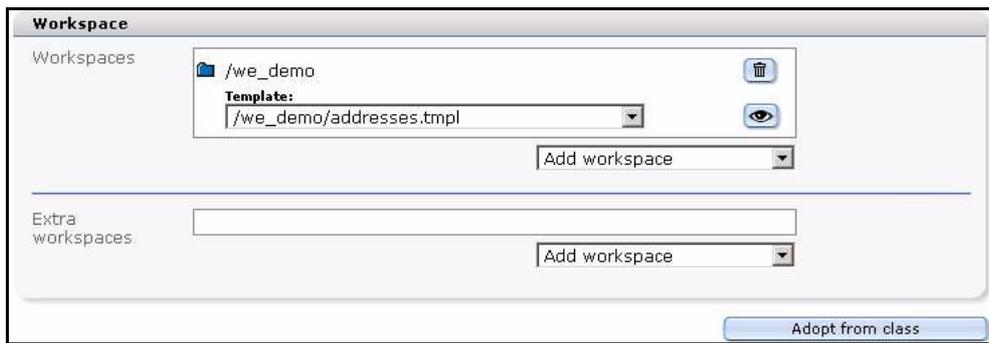
- 2 In the *Properties* view, enter a name for the object in the *Object name* field (see Figure 41).

Figure 41 Creating a new object

The screenshot shows a dialog box titled 'Object:'. It has several tabs: 'Properties', 'Edit', 'Workspace', 'Information', 'Preview', and 'Scheduler PRO'. The 'Properties' tab is active. Inside, there are three main sections: 'Path', 'Object name', and 'Class'. The 'Object name' field contains the text 'Baseball_field'. Below it, the 'Directory' field contains '/addresses' and has a 'Select' button to its right. At the bottom, the 'Class' dropdown menu is set to '/addresses'.

Note: If you selected the automatic naming feature when defining the object class, you do not have to enter a new name. The automatic naming feature automatically creates a name for your new object, ensuring that all objects within this class are named uniformly and in order. For details on the automatic naming feature, see Procedure 1, "Creating a new object class," Step 3 on page 41.

- 3 Click on the *More properties* icon.
 - a If you wish to associate the object class with an existing *Category*, select the *Add* button under the categories field.
A list of available categories appears in a dialog box from which you can choose a category.
 - b If you wish to define an owner, enter the name in the *Owners* area
The user name of the owner of the object appears in the Owners area. (This area is only available if the User Management or User Management PRO modules have been installed.) If a new object is created, the current user is the main owner.
 - c For users of the User Management and User Management PRO modules:
 - i If you wish to limit access to the object, you can activate the *Restrict access* function. There, you can select which users are allowed to access the object.
 - ii You can also activate the *Read only* checkbox that appears to the right of the user name option so that they can read the object but not change it.
 - d If you wish to create a long series of objects that are only slightly different from one another, use the *Copy object* function. The *Copy object* function allows you to import an existing object with all its settings. Thereafter you can make the necessary minor changes in the *Edit* view (see the following step) and then save the new object.
- 4 Save your entries by clicking the *Save* button.
- 5 Click on the *Edit* tab.
The Edit view opens.
- 6 Fill in the individual fields that appear in the *Edit* view for the object and save the new object.
Depending on the method used to save it, the object will then be displayed either in all the corresponding summaries (listviews) or in the detailed view.
- 7 Assign the workspace(s) for which the objects in this object class are to be visible.
The file manager opens in which you can select a directory (workspace), as seen in Figure 42.

Figure 42 Assigning a workspace to an object

Note: If you do not specify a workspace, you can choose from the buttons in the Display area whether or not to display an object. The object will always be displayed using a document (listview, detailed view).

8 Verify the details about your object by clicking on the *Information* tab.

9 Click on the *Save* button.

Your new object will appear in the explorer menu under its object class.

10 You have completed this procedure.

2.3 Using workspaces

The following examples show how you can assign different workspaces to selected objects.

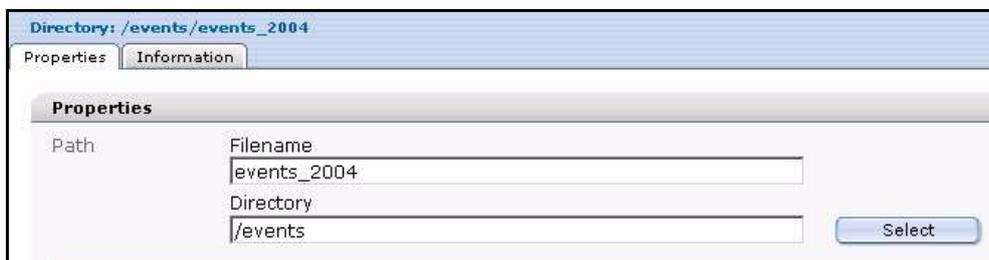
2.3.1 Assigning a new workspace

The purpose of this exercise is to show you how to create an object directory and an object, and to assign a new workspace to it.

Procedure 3 Assigning a new workspace, and deleting an old one

In the Objects view

1 Create a subdirectory with the name *events_2004* under the *events* class on the webEdition demo page by selecting *File > New > Directory > Object directory* (see Figure 43).

Figure 43 Creating a subclass

2 In the *Path* area, enter the name *events_2004* in the *Filename* field.

3 Click on the *Select* button adjacent to the *Directory* field.

The file manager opens, from which you can choose a directory.

4 Select the *events* directory and click the *OK* button.

5 Click the *Save* button at the bottom of the *Properties* view.

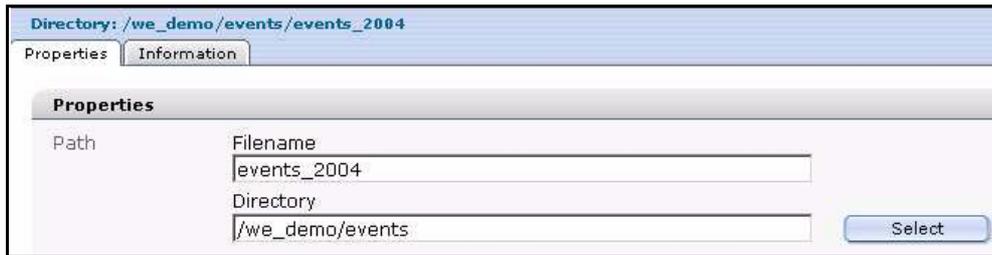
The system confirms that the new directory has been saved. After you close this dialog box, your new directory appears in the explorer menu as a subfolder under the events folder.

- 6 Create a subdirectory called `events_2004` under the *Documents* view by selecting *File > New > Directory > Documents directory*.

The system opens the *Documents Properties* view (see Figure 44).

- 7 Repeat Step 2 through Step 5.

Figure 44 New documents subdirectory



- 8 Create a new object called `2004_01_01_01` in the newly created subclass object folder `events_2004`. To do so, click *File > New > Object > events*, type `2004_01_01_01` in the *Object name* field, and choose the `events_2004` directory using the *Select* button adjacent to the *Directory* field.

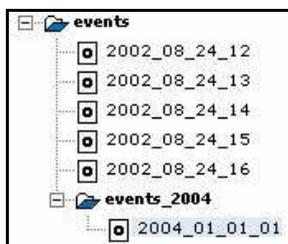
- 9 While still in the *Objects* view, click the *Edit* tab, and enter values into the required fields.

Note: The required fields are noted with an asterisk (*).

- 10 Save and publish the new object in the subclass object folder `events_2004`.

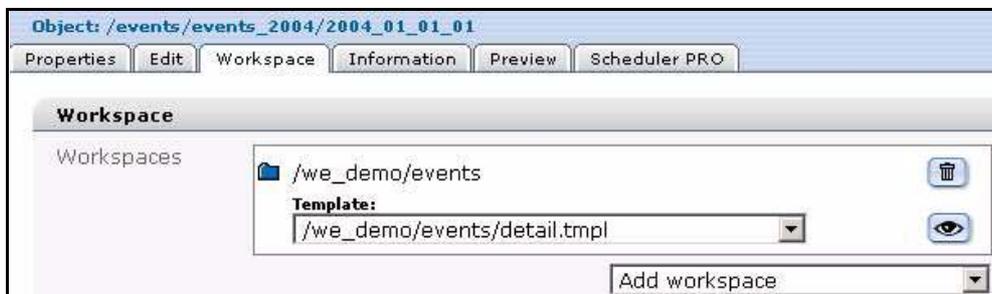
You will see the following display in the explorer menu:

Figure 45 New object



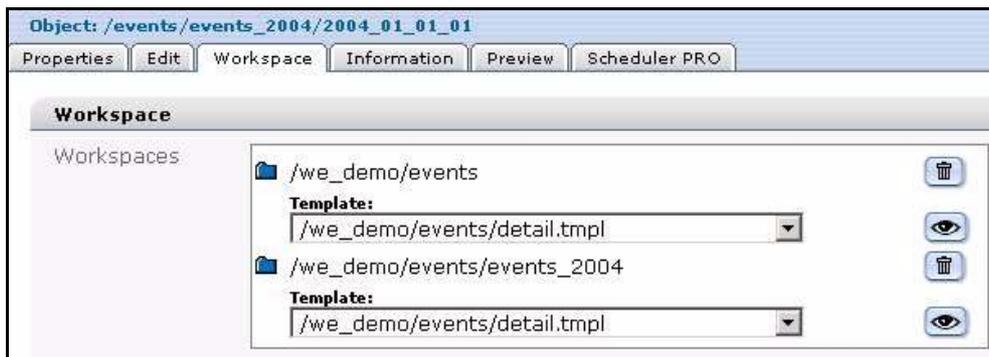
- 11 Click on the new object `2004_01_01_01` and select the *Workspace* view (see Figure 46).

Figure 46 Workspace for the new object



- 12 Add a new workspace `/we_demo/events/events_2004` from the *Add workspace* drop-down menu (see Figure 47). (This workspace is actually the new subdirectory that you created in the document view in Step 6 and Step 7 of this procedure).

Figure 47 Add a new workspace



- 13 Delete the old workspace `/we_demo_events` by clicking the trash can icon adjacent to the field, and click the Save button.

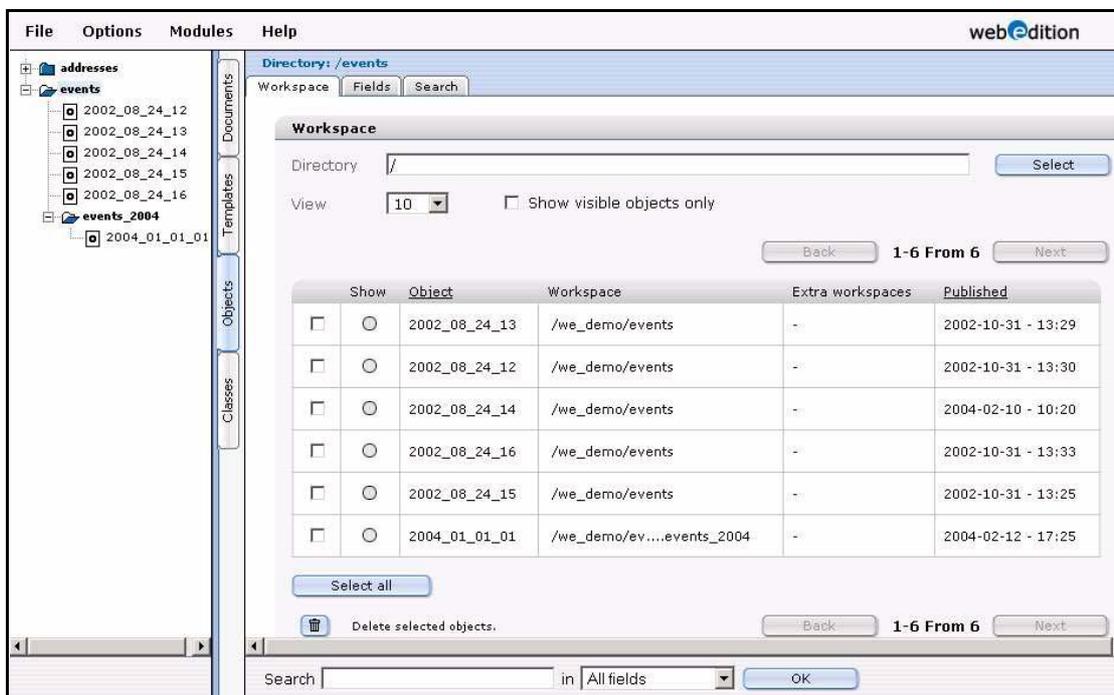
You are be left with the new workspace `/we_demo/events/events_2004`.

- 14 In the *Objects* view, click on the *events* class folder in the explorer menu to view an overall summary of all objects contained in the *events* class.

The Workspace view appears as seen in Figure 48, "Summary of all objects in the events class" on page 48.

Note: *If you have not selected a directory and are still in the root directory, all the circles in the Show column will be gray because none of the objects can be displayed in the root directory.*

Figure 48 Summary of all objects in the events class



- 15 In the *Workspace* view, choose the `/we_demo/events` directory by clicking the *Select* button adjacent to the *Directory* field.

The circles next to the object (except for the new one) in the Show column turn blue, indicating that the object can be displayed in the assigned workspace (see Figure 49). If you select the `/we_demo/events/events_2004` directory, the circle next to the new object will also become blue, indicating that it can be displayed in this workspace.

Figure 49 Display properties for objects in a workspace

Show	Object	Workspace
<input type="radio"/>	2002_08_24_13	/we_demo/events
<input type="radio"/>	2002_08_24_12	/we_demo/events
<input type="radio"/>	2002_08_24_14	/we_demo/events
<input type="radio"/>	2002_08_24_16	/we_demo/events
<input type="radio"/>	2002_08_24_15	/we_demo/events
<input type="radio"/>	2004_01_01_01	/we_demo/ev....events_2004

16 You can now view the object by clicking on it in the explorer menu, then clicking the *Preview* view.

Figure 50 Viewing an object with the *Preview* view

17 You have completed this procedure.

2.3.1.1 Assigning an extra workspace

The following procedure builds upon what was discussed in Section 2.3.1, "Assigning a new workspace" on page 46. In this procedure you will learn how to create extra workspaces for your objects.

Procedure 4 Assigning an extra workspace

In the Objects view

- 1 Click the new object 2004_01_01_01 in the *Object* view, that you created in Procedure 3, "Assigning a new workspace, and deleting an old one" on page 46.
- 2 Select the *Workspace* view.
- 3 In the *Extra workspaces* area, select the /we_demo/events from the *Add workspace* drop-down menu.
- 4 Activate the *Publish after saving* check box, and click the *Save* button.
- 5 In the *Objects* view, click on the events class directory.

A summary of all objects in the events class appears in which you see an entry in the Extra workspaces column in the row assigned to the object 2004_01_01_01 (see Figure 51):

Figure 51 Summary of objects

<input checked="" type="checkbox"/>	<input type="radio"/>	2004_01_01_01	/we_demo/ev....events_2004	<input checked="" type="checkbox"/>	/we_demo/events	2004-02-12 - 17:47
-------------------------------------	-----------------------	---------------	----------------------------	-------------------------------------	-----------------	--------------------

- 6 You can share the object for this workspace by activating the checkbox in the *Extra workspaces* column.

If this workspace contains a document with a listview of the "Events" objects, the new object will not be visible until this checkbox has been checked. The object will become visible once the checkbox has been checked.

- 7 You have completed this procedure.

3 Displaying objects and including objects in templates

In Chapter 2 we described how to create object classes, how to assign a template to a class, and how to create objects. In this chapter we shall discuss the template code and `we:tags` that you can use to include objects in your templates so they can be published on a web page.

The display of objects as a list or in a detailed display in the front end is always done by means of templates and documents. If you want to display stored objects, you will have to create a document based on a template that contains the corresponding listview or fields for a detailed display.

The following list draws upon what was treated in Chapter 2 and it outlines a suggested order for performing the procedures required to create *and* publish objects.

1. Create an object class
2. Assign a detail template to the class
3. Create a detail template (to display the object when linked to/from a listview)
4. Create an object
5. Create a template that can search for an object. (listview)
6. Create a web page from the template made in step 5

In this chapter we shall focus on Steps 3 and 5. For details on Step 6, see the *webEdition User Guide*.

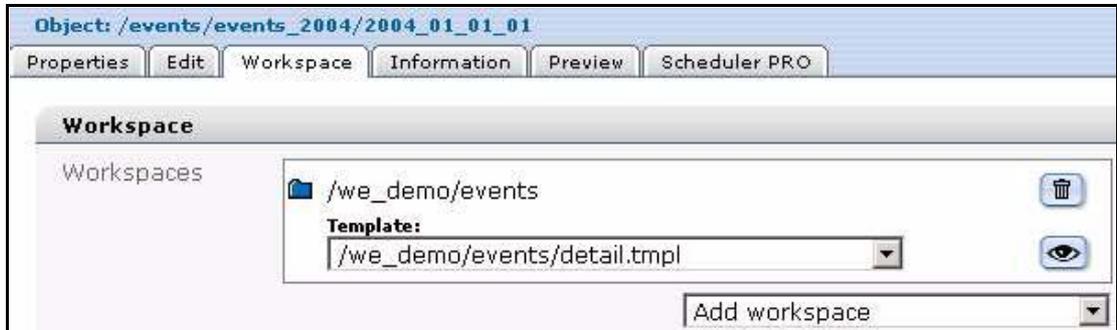
3.1 Displaying objects: an overview

Objects themselves do not contain any layout information or other display options, therefore, you must always link an object to a webEdition document. If you want a summary of the objects, you first must create a template that contains a corresponding listview. Thereafter you can use the template to create a document.

If, for example, you want to display an object in the "Events" workspace, you first must save a corresponding document (`summary.php`) with a listview of all objects of the "Events" class in a directory within this workspace. If the document (`summary.php`) is saved outside of this workspace, the listview will not find any objects.

To display an object in detail, you must link the object to an appropriate template. You can do so in the *Workspace* view. Once the workspace has been assigned, you can select the template for the detailed view of the object (see Figure 52).

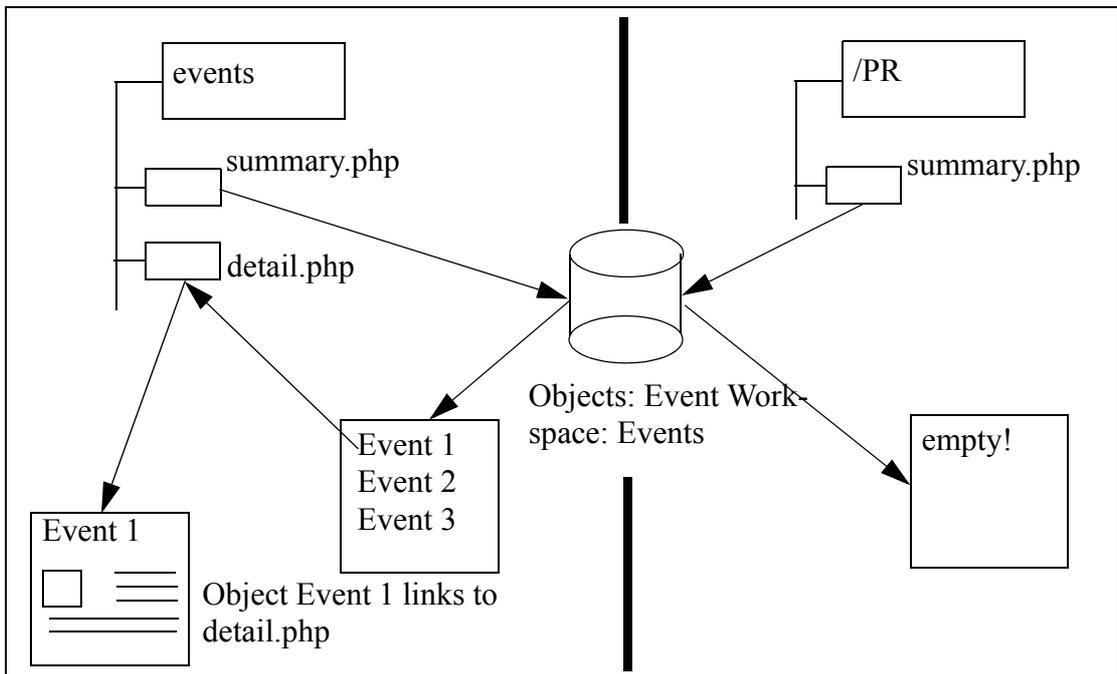
Figure 52 Selected template



Once you have selected a template, the object will always be displayed in the frontend with this template whenever the link is clicked in the listview. You can select a different template for each selected workspace.

Figure 53 is a schematic diagram of workspaces and assigned templates for objects. In this case, only the "Events" workspace has been assigned to the "Events" object class.

Figure 53 Diagram of workspaces and templates



For more information, see Section 3.4, "Displaying an object in a document using the `<we:listview>` tag" on page 54.

3.2 Database/Object Module we:tags

There are four tags that you can use for including objects in your templates:

- `<we:author>`
- `<we:ifNotObject>`
- `<we:ifObject>`
- `<we:object>...</we:object>`

The Database/Object Module also expands the capability of existing we:tags to include new attributes:

- `<we:field type="select">`
- `<we:input type="select">`
- `<we:listview type="search">`
- `<we:listview type="object">`
- `<we:var type="select">`

For information about these tags, see *The webEdition Tag Reference Manual*, which is available on-line at the following URL: <http://www.webedition.biz>

3.3 Creating an object summary

webEdition allows you to create object summaries by using the `<we:listview>` tag. After installing the Database/Object Module, the *object* attribute can assist you in creating summaries of objects. The following code example shows you how to use the `<we:listview type = object>` tag. The template output appears in Figure 54, "Template output" on page 54.

```
<we:listview type="object" classid="2" rows="4" order="EventDate">
  <we:repeat>
    <p>
      <we:field type="date" name="EventDate"> -
      <we:field name="EventName" hyperlink="true"/>
      <br>
      <we:field name="EventDescription">
    </p>
  </we:repeat>
</we:listview>
```

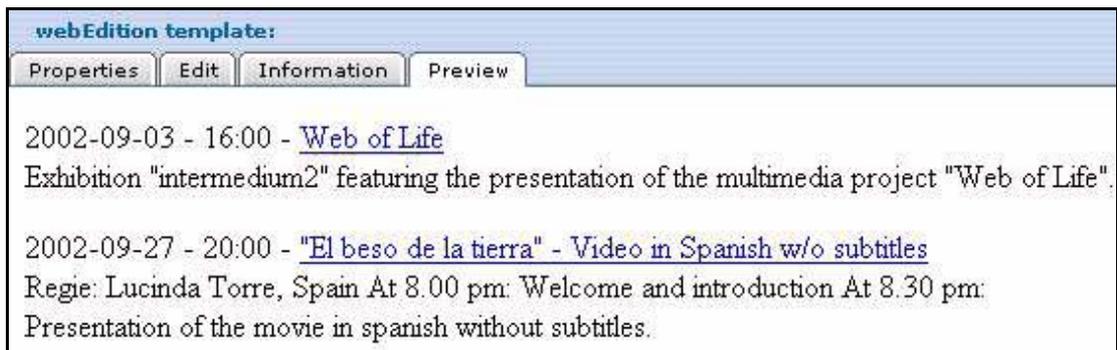
This code generates a summary of all objects from the "Events" object class. The "Events" object class has the ID=2. The list will be sorted by the "EventDate" class field.

The list will display:

- the "EventDate" field
- a hyphen "-"
- the "EventName" field
- the "EventDescription" field

The "Event Name" (`hyperlink="true"`) field is linked to the template that has been assigned to the object class in the *Workspace* view.

Figure 54 Template output

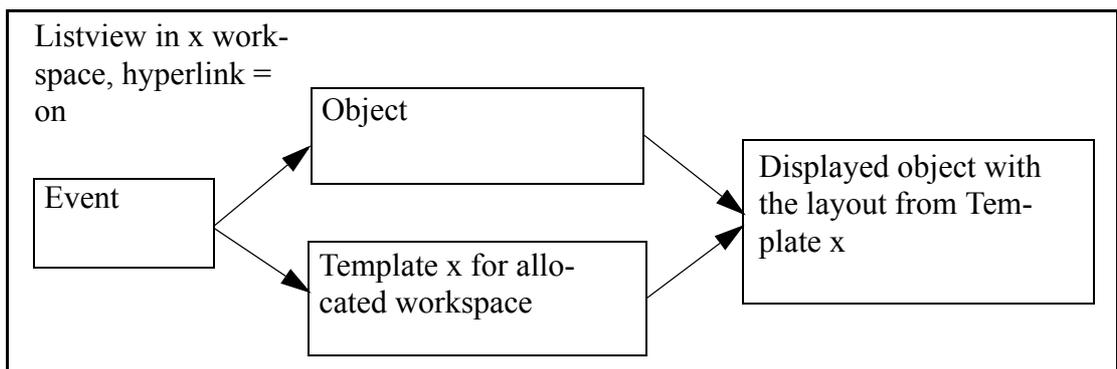


3.4 Displaying an object in a document using the <we:listview> tag

To display an individual object in a document, you first have to assign a template to the object class. This can be done when defining the object classes in the *Workspace* view in *Edit* mode. As described in Section 1.7.2, "Displaying objects in the front end" on page 38, a workspace is allocated to an object class in which the objects can be displayed. A default template for the detailed display of the object is also allocated. If you have assigned several workspaces to an object class, you can also assign several default templates so that the object can have different templates, that is, layouts or levels of information, depending on which workspace it is displayed in.

The link is always created in conjunction with a listview as seen in the sample code in Section 3.3, "Creating an object summary" on page 53.

Because of the `hyperlink="true"` attribute, the Event Name is defined as a link that points to the default template defined in the *Workspace* view of the object class for the workspace where the document with the listview has been stored. It also points to the object listed in the listview. Clicking on this link will combine and display the object data and the template as shown in Figure 55.

Figure 55 Example: `hyperlink="true"`

The variable of the name attribute in the `we:tag <we:field>` must be absolutely identical with the desired field name of the object. (This tag is case-sensitive.)

3.5 Displaying an object in a document using the <we:object> tag

You can include individual objects directly in documents without listing them in a listview. This is advisable, for instance, if you have an "Events" template in which your

editors manually enter the event details and then insert the address where the event is to be held. You must insert the we:tag `<we:object>` below the text entry fields for the event. With this tag, you can give editors the ability to select certain objects in a class. You must always specify an object class. Defining both the class and the specific object will cause this object to be shown in this document. This is done using the we:tags `<we:object>` and `<we:field>`. For more information, see the descriptions of the `<we:object>` and `<we:field>` tags in *The webEdition Tag Reference Manual*.

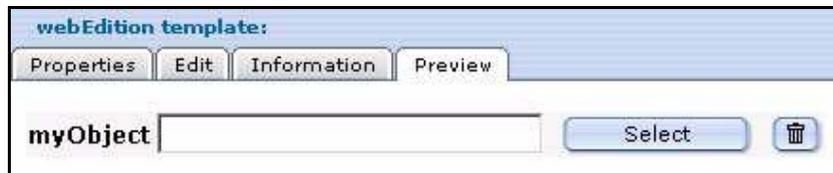
```
<we:object name="myObject" classid="77" id="324">
  <p>
    <we:field name="name1" hyperlink="true"/><br>
    <we:field name="text"/>
  </p>
</we:object>
```

This code always displays the same object in the class that possesses the `classid="77"` and the object that has the `id="324"` without permitting the editor to change anything.

```
<we:object name="myObject" classid="77">
  <p>
    <we:field name="name1" hyperlink="true"/><br>
    <we:field name="text"/>
  </p>
</we:object>
```

In this case, the editor can select any object from all objects that exist under the object class that has the `classid="77"`. The editor will see the following display in the front end (see Figure 56):

Figure 56 Displaying an object



Clicking the select button will cause the file manager to open so that the editor can select an object of `classid="77"`.

The `<we:field>` tag can be used to define which object field should be displayed. The variable of the name attribute in the we:tag `<we:field>` must be absolutely identical to the desired field name of the object. (This tag is case-sensitive.)

Index

A

Audience 11
 audience profile 11
 automatic naming
 by day 31
 by hours 31
 by id 31
 by month 31
 by text 32
 by year 31
 unique 31
 using 30

C

characters permitted in Name field 43
 class summary
 Fields view 25
 Search view 27
 Workspace view 24
 Classes tab
 and the classes explorer menu 27
 Edit view 28
 Information view 30
 Properties view 28
 using 27
 Workspace view 29
 creating objects and classes
 permissions 38
 Customer service 14

D

Database/Object Module
 installation 17
 intended users 15
 interaction with other modules
 Scheduler PRO Module 17
 User Management Module 17
 Workflow Module 17
 menus 18
 permissions in 38

suggested uses 16
 uses for 16
 we:tags 53

E

explorer menu
 viewing object classes in 19
 viewing objects in 19

F

Feldtypen
 Textarea 33
 field types
 about 32
 Binary 37
 Date 33
 Float 36
 Href 36
 Image 35
 Integer 35
 Link 36
 Object 36
 Select 32
 Text input 33
 fields
 combining 37
 deleting 37
 sorting 37

I

icons
 object 19
 object class 27
 Information view described 22

O

Object class summaries 24
 Object classes
 creating 41

- object classes
 - defined 16
- Object field type 36
- Objects
 - Edit view described 21
 - Properties view described 20
 - Workspace view described 21
- objects
 - allocating a template for 44
 - copying 45
 - creating 44
 - defined 15
 - displaying 38
 - overview 51
 - process for 51
 - saving 23
- Objects tab
 - and the objects explorer menu 19
 - using 19
 - views available with 20
- P**
- Precautionary messages
 - about 12
 - Attention boxes 12
 - Caution boxes 12
- Properties view
 - for classes 28
 - for objects 20
- R**
- reference documentation 11
- Restricting user access 42
- S**
- search operators 27
- T**
- templates
 - linking objects to 52
 - we:tags for the Database/Object Module 53
- Typographical conventions 13
- U**
- User Management Module
 - interaction with the Database/Object Module 42, 45
- W**
- we:tags
 - attribute values 13
 - we:author 53
 - we:field type="select" 53
 - we:ifNotObject 53
 - we:ifObject 53
 - we:input type="select" 53
 - we:listview
 - used to create object summaries 53
 - used to display an object 54
 - we:listview type="object" 53
 - we:listview type="search" 53
 - we:object 53
 - used to display an object 54
 - we:var type="select" 53
- webEdition
 - documentation suite 11
 - on-line documentation 11
 - we:tags 13
- webEdition documentation
 - Customer documentation suite 11
 - on the World Wide Web 11
 - version and issue 13
- Workflow Module
 - interaction with Database/Object Module 23
- workspace
 - defined 21
 - view 22
- workspaces
 - assigning for an object class 43

Database/Object Module

User Guide

Standard 03.00
09 February 2004
Printed in Germany

© 2004 webEdition Software GmbH
All rights reserved.



webEdition Software GmbH
Karlsburgstr. 2
76227 Karlsruhe
Germany

Internet: <http://www.webedition.biz>
E-mail: info@webedition.de

web **edition**
Software GmbH