

Blue Chameleon CMS (Other elements)

*For links outside this document,
download the relevant chapter or the*

Blue Chameleon Content Management System full documentation.

May 8, 2012

Contents

1	Configuring and administration	13
1.1	System interface	13
1.2	Users and usergroups	14
1.2.1	Users	14
1.2.1.1	User groups that a user belongs to	15
1.2.1.2	User management through time	16
1.2.2	User groups	16
1.2.3	User groups access rights	16
1.3	Blue Chameleon Content Management System user rights	17
1.3.1	Rights on headlines	17
1.3.2	Rights on Articles	18
1.3.3	Rights on page objects	18
1.3.4	Rights on Images	18
1.3.5	Rights on Search	18
1.3.6	Rights on URLs	18
1.3.7	Rights on columns	19
1.3.8	Rights on publishing	19
1.3.9	Rights on Data Base Publishing	20
1.3.10	Rights on Editions	20
1.3.11	Rights on Background Frames	20
1.3.12	Rights on Downloads	20
1.3.13	Rights on Site management	21
1.3.14	Rights on Article Models	21
1.3.15	Rights on Headline Models	21
1.3.16	Rights on Notice boards	21
1.3.17	Rights on System log	22
1.3.18	Rights on Scripts	22
1.3.19	Administrator rights	22
1.3.20	Rights on Hermetic groups	22
1.3.21	Information level	23
1.3.22	Rights on Books	23
1.3.23	Rights on Book Models	23
1.3.24	Java option menu	23
1.3.25	Connection speed	23
1.3.26	Rights on User management	24
1.3.27	Rights on Menus, Menu Templates	24

1.3.28	Rights on Packages	24
1.3.29	Rights on Forms	24
1.4	Columns	25
1.4.1	Publisher and objects	25
1.4.2	Creating columns	25
1.4.3	Managing columns	26
1.4.3.1	Changing columns	26
1.4.3.2	Updating columns	26
1.4.3.3	Backups	27
1.4.4	Blue Chameleon directory structure	28
1.4.4.1	Virtual columns	28
1.5	Publisher settings	28
1.5.1	File types	29
1.6	FTP profiles	30
2	Composing content	33
2.1	Principles	33
2.2	Headlines models	33
2.2.1	Creating a new headline model	34
2.2.2	Updating a headline model	35
2.2.3	Contents of a headline model	36
2.2.3.1	Enriching a headline model with frames	37
2.2.3.1.1	Frame templates used by a headline model	39
2.2.3.1.2	Updating frame number	40
2.2.3.2	Meta Tags	40
2.3	Frame templates	41
2.3.1	Creating and updating frame templates	41
2.3.2	"List" frames	42
2.3.3	"Leaf" frames	43
2.3.4	Frame style	44
2.4	Headlines : creation and editing	45
2.4.1	Editing a headline : a basic example	45
2.4.2	Editing a leaf frame	46
2.4.3	Editing a list frame	47
2.4.3.1	Example for a leaf frame as used by a list	48
2.5	Layouts	50
2.6	Articles and article models	50
2.7	Managing pages	50
2.7.1	Publishing pages	50
2.7.1.1	Multiple publish	51
2.7.1.2	Scheduled publishing	52
2.7.2	Properties of headlines	53
2.7.3	Other management options for pages	54

3	Object management	57
3.1	Images	57
3.1.1	Uploading an image	57
3.1.2	Image folders	59
3.1.3	Managing images	59
3.1.4	Integrating images	59
3.2	Links	60
3.2.1	Creating a URL link	61
3.2.2	Managing URL links	62
3.2.3	Integrating links	62
3.3	Scripts	64
3.3.1	Creating a script - Management	64
3.3.2	Integrating a frame script	65
3.3.3	Integrating a link style script	66
3.4	Downloads	67
3.4.1	Creating a download	67
3.4.2	Managing downloads	68
3.4.3	Integrating downloads	68
3.5	Menus	69
3.5.1	Menu scripts	69
3.5.2	Menu models	71
3.5.3	Creation of a menu	71
3.5.3.1	Filling of a menu	72
3.5.3.2	Menu result and integration	74
3.5.4	More complex menus	75
3.5.4.1	A menu script that handles children menu items	75
3.5.4.2	A menu with children menu items	75
3.6	Forms	78
3.6.1	Form models	78
3.6.1.1	Form model management - parameters	79
3.6.2	Defining forms	79
3.6.2.1	Updating a form	80
3.6.3	Composing a form	80
3.6.3.1	Editing a form page	81
3.6.3.2	Form page element types	82
3.6.3.3	Editing a form page element	83
3.6.3.4	After form is submitted	84
3.6.3.5	A form result	84
3.6.3.6	A form with several pages	84
3.6.4	Integrating a form	85
3.7	Variable files	86
3.7.1	Example of varfile contents	86
3.7.2	Varfiles as variables	87

4	Other elements	89
4.1	Site management	89
4.1.1	Uploading files	89
4.1.2	Creating directories	90
4.1.3	Managing files and directories	91
4.1.4	Executing SQL	91
4.1.5	Export and Import	93
	4.1.5.1 Generating export files - importing them	93
	4.1.5.2 Export file contents	94
4.2	Model defaults	96
4.2.1	Model defaults and headlines	96
4.3	Books	97
4.3.1	Book models	97
4.3.2	Creating books	97
4.3.3	Modifying a book	99
4.4	Notice boards	100
4.4.1	Creating and managing notice boards	100
4.4.2	Association with a column	100
4.4.3	Notice board entries	101
4.5	Editions	102
4.5.1	General edition management	103
4.5.2	Publishing editions	104
4.6	Options	104
4.6.1	Version	104
4.6.2	System Logs	105
4.6.3	Expiration report	106
4.7	Searching for content	106
4.8	Management of objects	108
5	Packages	109
5.1	Mandatory scripts	109
5.1.1	PackageInstall.phs	109
5.1.2	PackageGroupModify.phs, PackageGroupModify1.phs	110
	5.1.2.1 Result : package group rights	110
5.1.3	PackageMain.phs	113
	5.1.3.1 Result : package main page	114
5.1.4	PackageUninstall.phs	116
5.2	Package general management	117
5.2.1	Installing a package	117
5.2.2	Updating a package	117
5.2.3	Uninstalling a package	118
5.3	Using object types and objects to compose headlines	118
5.3.1	Enabling a leaf frame to handle packages and an object	119
5.3.2	PackageObjTypeList.phs	120
5.3.3	PackageObjectList.phs	122
5.3.4	PackageObjectName.phs	122

5.3.5	PackageObjectView.phs	124
5.4	Using attributes	125
5.4.1	Enabling a frame to handle attributes	125
5.4.2	Scripts for attribute	126
5.5	Featuring multiple objects on the same headline	127
5.5.1	A leaf frame that handles multiple objects	128
5.5.2	Packages : defining more of them	130
5.5.3	Object and attribute scripts for this case (guidelines)	130
5.5.3.1	The PackageObjectView.phs script for this example	132
6	Annex	135
6.1	Leaf frame variables	135
6.1.1	"Table" variables	137
6.1.2	"Select" variables	137
6.1.3	"SQL select" variables	138
6.2	System script glossary	138
6.2.1	Headline model scripts	138
6.2.2	Leaf frame model scripts	139
6.3	Icon glossary	141
6.4	Form page elements : detailed	143
6.4.1	"Active" elements	143
6.4.1.1	Checkboxes	143
6.4.1.2	Radiobuttons	144
6.4.1.3	Listboxes	144
6.4.1.4	Input field and text area	145
6.4.1.5	Form label - 'with check box'	146
6.4.2	"Passive" elements	146
6.4.2.1	Hidden field	146
6.4.2.2	Form image	146
6.4.2.3	Form label - 'normal'	146
6.4.2.4	Form separator	147

List of Figures

1.1	The Publisher's main page.	13
1.2	Defining a new user.	15
1.3	Groups a user belongs to.	15
1.4	Defining a user group.	16
1.5	The most important access rights as defined for a user group.	17
1.6	Creating a new column.	26
1.7	Updating a column.	27
1.8	The properties of your Publisher.	29
1.9	Defining a FTP profile.	31
2.1	Creating a headline model.	34
2.2	Here, a headline model can be updated and its children frame templates be chosen.	36
2.3	Thanks to this, the pages created on this model will have meta-tags called Author, LastUpdate and a Page Title.	41
2.4	Creating a new headline.	45
2.5	First edit ; as both frames use only one template ("Basic text frame" and "Side Menu List"), so clicking on the icon shows them immediately.	46
2.6	Filling in with information the leaf's various variables.	47
2.7	A preview of the page.	48
2.8	Populating a list frame.	49
2.9	Publishing several headlines in one single time.	51
2.10	Configuring and viewing data related to headline "Home Page".	53
2.11	Managing headlines in a file system way.	55
3.1	Uploading an image.	58
3.2	Creating a folder for images.	59
3.3	From here, all images and folders can be globally managed.	60
3.4	Integrating an uploaded image.	61
3.5	Creating a link.	62
3.6	Putting a previously-defined URL link.	63
3.7	Creating a script to be used in a leaf frame.	65
3.8	Uploading a file that will be used for downloading purposes.	67
3.9	Integrating a download, with a 'download' variable.	69
3.10	Creating a menu model.	71
3.11	Creating a menu "Basic top menu" based upon model 'Top Menu'.	72
3.12	Creating a form model.	78

3.13	Adding parameters to a form model.	79
3.14	Creating a form based on the example model.	79
4.1	Uploading a file to be used in headline contents.	90
4.2	Creating a new directory, directly under the «#SQLWWWHOME#» root.	91
4.3	Editing a text-based file.	92
4.4	Outputting the contents of a table.	93
4.5	Exporting the current column.	94
4.6	An example of export file contents.	95
4.7	Creating a model default for a headline model.	96
4.8	Creating a book model.	98
4.9	Creating a book.	98
4.10	From here, book elements can be managed.	100
4.11	Creating a general notice board.	101
4.12	Manually adding two entries to this notice board.	102
4.13	Checking one's own activity for last week.	106
4.14	Performing a search on objects which publishing name begin by 'Josh'.	107
4.15	From here, any object can be managed.	108
5.1	Modifying this package's rights for a particular user group.	110
5.2	The main page for package Festivals.	114
5.3	The first install of the package "Festivals".	117
5.4	The process of selecting an object type and an object for a leaf frame that is package-enabled.	119
5.5	A selection of object types.	121
5.6	Finally choosing the object.	123
5.7	A page design that calls for multiple objects to be featured.	128
5.8	The 3-object frame after all relevant choices of objects and attributes have been made.	132
6.1	Composing a checkbox selection for a form.	144

List of Tables

5.1	Scripts, functions and variables as used for attribute setting	126
6.1	Leaf frame variables	136
6.2	Leaf frame variables (continued)	137
6.3	System scripts for headline, list and container models	139
6.4	System scripts for leaves	140
6.5	System scripts for leaves (continued)	141
6.6	Blue Chameleon Content Management icons	142
6.7	Blue Chameleon Content Management icons (continued)	143

Chapter 4

Other elements

This Chapter introduces components that, while quite secondary to Blue Chameleon Content Management System use, are nevertheless useful.

4.1 Site management



User group rights for Site Management are detailed at 1.3.13.

The previous ways of uploading data - images (3.1), links (3.2) and downloads (3.4) - showed a dedicated interface with the uploaded data being integrated on a leaf frame, through a variable.

Now, it is time to focus on files that can easily be used when composing a page's content, such as a Banner image (as in Code 1), a `css` style... in other words, files to be used in **contents** as accessed by [Headlines](#) / [Modify](#). They need to be easily linked to, with HTML code.

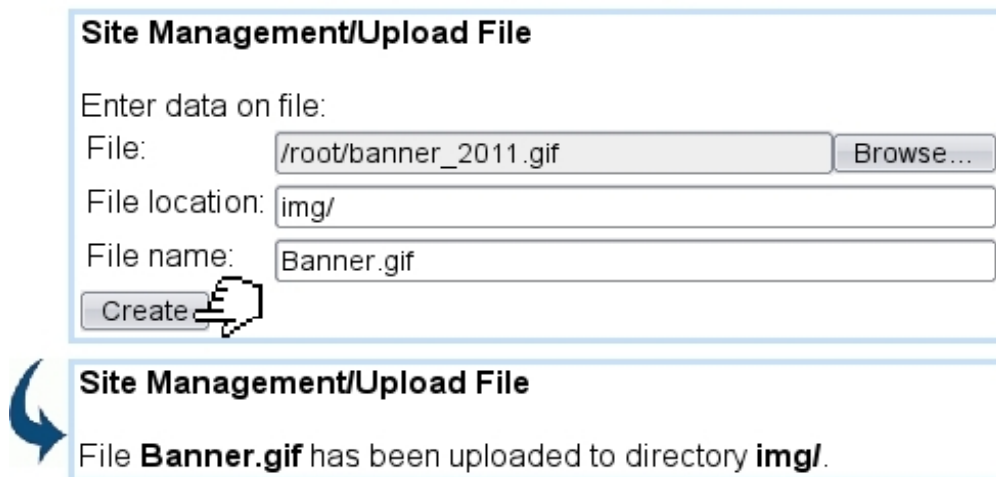
The paths to upload files, in the ways previously described, were consisting in columns, mainly `opsmain`. The uploading of files through [Site management](#) happens at the root described by the system variable `<#SQLWWWHOME#>`. From here, folders can also be created (4.1.2).

4.1.1 Uploading files

Fig.4.1 shows how the image `Banner.gif` is uploaded onto the already-existing `img/` folder. There are given :

- the directory under which it will be uploaded e.g. `img/`, `css/`,... (this directory must already exist, otherwise it has to be created - see below) ;
- the full name under which it will be used (which can be different from the original file name) ;

[Site management](#) / [Upload file](#) :



The image shows two screenshots of a web interface. The top screenshot is a form titled "Site Management/Upload File". It contains the following fields and buttons: "Enter data on file:" label, "File:" input with the value "/root/banner_2011.gif" and a "Browse..." button, "File location:" input with the value "img/", "File name:" input with the value "Banner.gif", and a "Create" button with a mouse cursor over it. The bottom screenshot is a confirmation message titled "Site Management/Upload File" with a blue arrow icon on the left. The message text is "File **Banner.gif** has been uploaded to directory **img/**."

Figure 4.1: Uploading a file to be used in headline contents.

When uploaded, files can then be used as in the following headline (or frame) sample, never forgetting the «#SQLWWWHOME#» root :

[Headline models](#) / [Modify](#) / [Some headline](#) :

```
<HTML>
<HEAD>
<LINK HREF="«#SQLWWWHOME#»/css/MyStyle.css" REL="stylesheet" TYPE="text/css"
MEDIA="screen">
</HEAD>
<BODY>

«INCLUDE design.phs»

<IMG SRC="«#SQLWWWHOME#»/img/Banner.gif">

(...) <SCRIPT type="text/javascript"
src="«#SQLWWWHOME#»/js/Calculation.js"></SCRIPT>
(...)
```

4.1.2 Creating directories

Directories can be created to sort better pictures, styles,... to be included in headline and frame templates' contents. Fig.4.2 shows how a new directory is created.

Blue Chameleon Content Management System already has the directories `img/`, `js/` and `css/`. To create a sub-directory under an already-existing directory, the full path must be given :

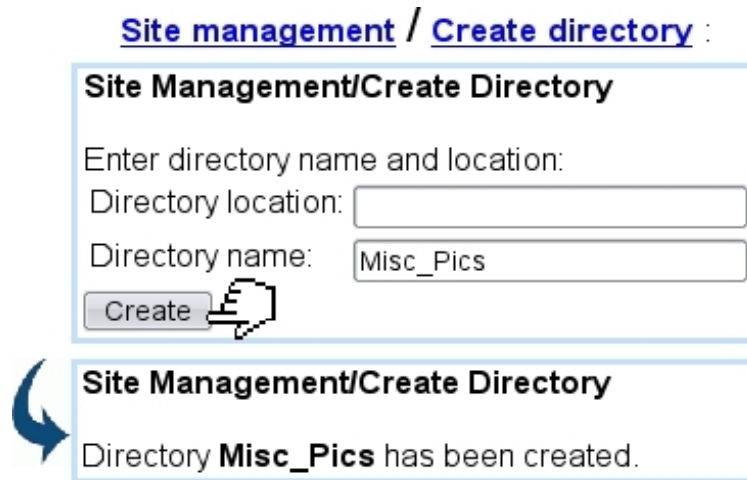
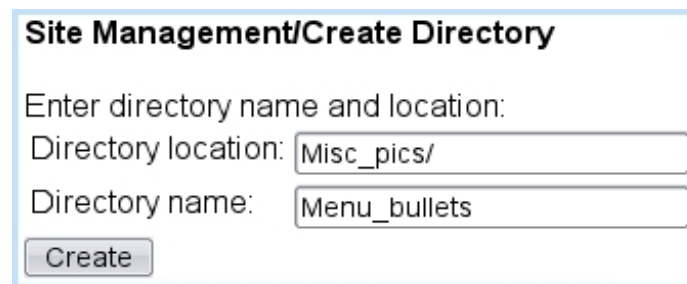


Figure 4.2: Creating a new directory, directly under the «#SQLWWWHOME#» root.



4.1.3 Managing files and directories

Text-based files such as scripts, css,... can be edited and saved as featured in Fig.4.3.

It is also possible to rename files with the eponymous link : file will be identified as in Fig.4.3 and a new name will be inputed for it.

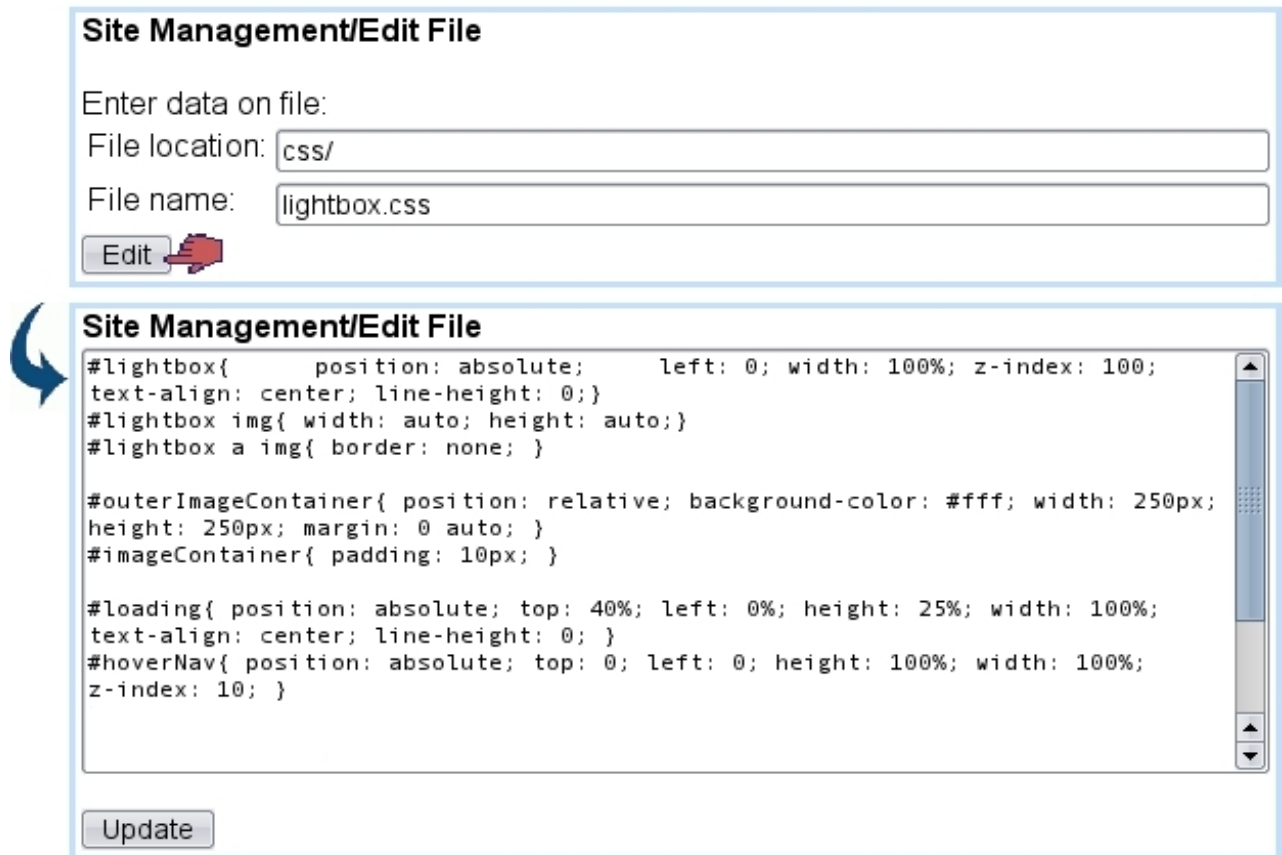
Deletions of files and directories are processed with the relevant links, with the same identification needed (location and name of file or directory).

4.1.4 Executing SQL

Executing SQL queries, an example of which is shown at Fig.4.4, is mainly important for Packages (5), where you have created your own tables (5.1.1). According to the nature of the query (SELECT or an other), a different button is clicked.

An example of table creation :

[Site management](#) / [Edit file](#) :



The screenshot shows two panels of the 'Site Management/Edit File' interface. The top panel is the input form, and the bottom panel is the code editor. A blue arrow points from the 'Edit' button in the top panel to the code editor in the bottom panel.

Site Management/Edit File

Enter data on file:

File location:

File name:

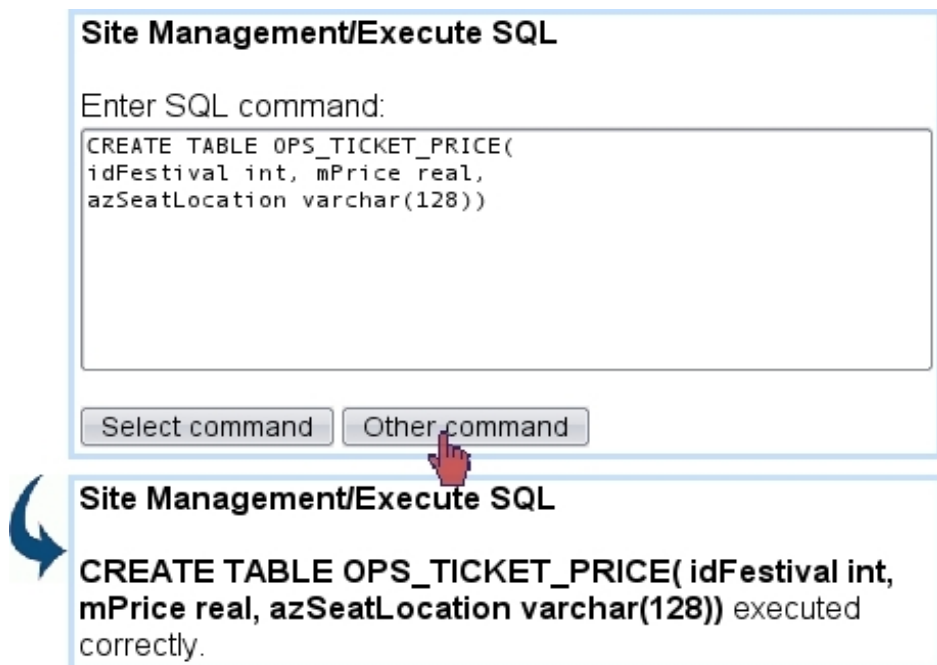
Site Management/Edit File

```
#lightbox{ position: absolute; left: 0; width: 100%; z-index: 100;
text-align: center; line-height: 0;}
#lightbox img{ width: auto; height: auto;}
#lightbox a img{ border: none; }

#outerImageContainer{ position: relative; background-color: #fff; width: 250px;
height: 250px; margin: 0 auto; }
#imageContainer{ padding: 10px; }

#loading{ position: absolute; top: 40%; left: 0%; height: 25%; width: 100%;
text-align: center; line-height: 0; }
#hoverNav{ position: absolute; top: 0; left: 0; height: 100%; width: 100%;
z-index: 10; }
```

Figure 4.3: Editing a text-based file.



The screenshot shows two panels of the 'Site Management/Execute SQL' interface. The top panel is the input form, and the bottom panel is the output display. A blue arrow points from the 'Other command' button in the top panel to the output display in the bottom panel.

Site Management/Execute SQL

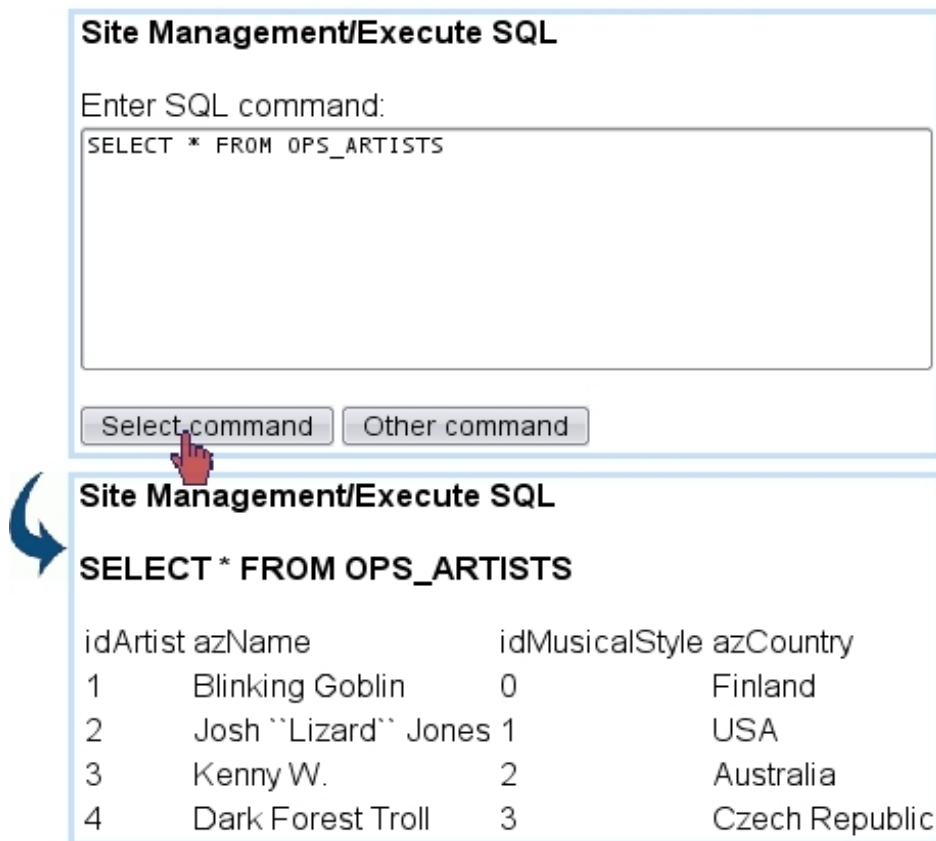
Enter SQL command:

```
CREATE TABLE OPS_TICKET_PRICE(
idFestival int, mPrice real,
azSeatLocation varchar(128))
```

Site Management/Execute SQL

CREATE TABLE OPS_TICKET_PRICE(idFestival int, mPrice real, azSeatLocation varchar(128)) executed correctly.

Site management / Execute SQL :



The screenshot shows a web interface for executing SQL commands. The top panel, titled 'Site Management/Execute SQL', contains a text input field with the command 'SELECT * FROM OPS_ARTISTS'. Below the input field are two buttons: 'Select command' and 'Other command'. A red hand cursor is pointing at the 'Select command' button. A blue arrow points from this button to the bottom panel, which also displays the command 'SELECT * FROM OPS_ARTISTS' and the resulting table output.

idArtist	azName	idMusicalStyle	azCountry
1	Blinking Goblin	0	Finland
2	Josh ``Lizard`` Jones	1	USA
3	Kenny W.	2	Australia
4	Dark Forest Troll	3	Czech Republic

Figure 4.4: Outputting the contents of a table.

4.1.5 Export and Import

The Publisher in general and its constituting columns in particular can be summed up in a formatted file that describe their contents with the help of opening and closing tags, holding with them data.

The aim of those files (which are `.txt` files) is to easily **export** the data contained in a column in a form that can be **imported** by another publisher, or even from a column of the publisher to another column.

4.1.5.1 Generating export files - importing them

Fig.4.5 shows how an `export.txt` file is generated.

There, for export purposes, two eponymous buttons serve as to either export the current column or the whole publisher. The file can then be seen through Blue Chameleon Content Management system via the [Edit file](#) link (4.1.3), leaving the 'File location:' blank as it is situated in the Publisher directory.

Similarly, an import of a well-formatted file can be done through the dedicated button, provided that file had been uploaded (4.1.1) under the Publisher root, i.e. with leaving



Figure 4.5: Exporting the current column.

again the 'File location:' blank.

4.1.5.2 Export file contents

Fig.4.6 shows an excerpt of the contents of an `export.txt` file.

There, it can be seen that it is organized along those rules :

- `<HEADLINE>...</HEADLINE>` tags embed the whole contents of a headline, of which basic information (its identifier, its related `COLUMN`, its `NAME`) is first given ;
- contents then consist in the file information this headline is based on (2.7.2), between `<FILE>...</FILE>`: name of the `.htm` file (`DEFFILE`) as well as the publication name (`FIXEDFILE`) ;
- then, more information on the headline itself :
 - the published status (`PUBLISHED_FLAG`) ;
 - if it belongs to an `EDITION` (4.5) ;
 - its `STYLE` ;
 - the identifier of the headline `MODEL` is it based on ;
 - the date of last publication (`PUBLISHED_DATE`) given as a Unix c-day ;
 - whether it has been given an `EXPIRY` date (2.7.2).
- `<FRAMESET>...</FRAMESET>` tags then embrace the headline's contents, i.e. the frame(s) the headline is made of, each between `<FRAME>...</FRAME>` tags ;
- within those, general information upon the frame is given :
 - its `INDEX`, i.e. the value of the `_FrameIndex` variable it is called by (2.2.3.1) ;
 - what its style (2.3.4) is, i.e. whether it is `SHOWN`, has a `BORDER` and has extended edition (`EXTEDITION`) ;
 - the identifier of the frame `MODEL`.

export.txt :

```
...
<HEADLINE>
NEW=1
COLUMN=1
ID=15
NAME=Home Page
<FILE>
DEFFILE=20110323143529_16608.htm
FIXEDFILE=Home_Page.htm
</FILE>
PUBLISHED_FLAG=0
EDITION=0
STYLE=0000
MODEL=44
PUBLISHED_DATE=40638
EXPIRY=0
<FRAMESET>
<FRAME>
INDEX=1
SHOWN=1
BORDER=1
EXTEDITION=1
MODEL=49
<VARFILE>
NAME=20110323145244_16737.var
<VAR>
NAME=_Text
VALUE=Blah
</VAR>
</VARFILE>
</FRAME>
...
<FRAME>
...
</FRAME>
</FRAMESET>
</HEADLINE>
...
```

Figure 4.6: An example of export file contents.

- then, in between `<VARFILE>...</VARFILE>` tags, the various variables for this frame are to be given, after the `NAME` of the related varfile is given. Variables are given as `NAME` and `VALUE` couples, in between `<VAR>...</VAR>` tags (see 3.7.1 for a detailed example of a varfile) ;
- as a frame contains only one varfile, the closing `</VARFILE>` tag is followed by the `</FRAME>` tag ;
- as a headline may contain several frames, the last of the closing `</FRAME>` tag(s) is followed by the closing `</FRAMESET>` tag, which is followed by the closing `</HEADLINE>` tag ;
- there are as many `<HEADLINE>...</HEADLINE>` tags as there are headlines in the

current column or Publisher.

4.2 Model defaults

A model default can be defined for Form models (3.6.1) and headline models (2.2).

At Fig.4.7, the creation of a model default aimed at headline model "Bk background" is illustrated.

Model Defaults / Create :

Model Defaults/Create

Enter Data on Model Defaults:

Description:

Image:

You must select an image if you are defining a headline model default.
You cannot select an image if you are defining a form model default.

Model: You must must choose one model, either a headline or a form model.

Form: ▾

Headline: ▾




Figure 4.7: Creating a model default for a headline model.

4.2.1 Model defaults and headlines

This model default can be afterwards associated with a headline, picked amongst a list of headlines based upon the headline model that was used.

Here, for model default "Bk model default bkgnd", using headline model "Bk background", we use a headline ("bk mod def headline") that was defined using that headline model. This setting can be done right after model default creation, or modification :

Model Defaults / Modify :

Model Defaults/Modify

Description:

Image:

Headline:

[Click here](#) to limit the usage of the model default to certain columns of the publisher.



4.3 Books

4.3.1 Book models



User group rights for managing Book Models are detailed at 1.3.23.



To be successfully created, book models necessitate the following to be defined :

- *'book' headline models (2.2.1) ;*
- *and fully-configured model defaults (4.2) for those headline models.*

Creating a new book model is featured at Fig.4.8.

There, apart from the name, picture of the book model and whether it is to be used in any column, book-type headline models are selected for the background and foreground.


After creation, a book model can be updated (name, picture, background/foreground models) or removed via [Model Defaults / Modify Book Model](#), [Delete Book Model](#).

4.3.2 Creating books



User group rights for managing Books are detailed at 1.3.22.

Books / Create Book Model :



Book/Create Book Model
Enter Data on Book Model:
Description:
Image:
Background Model:
Foreground Model:
 use book model in any column of the current Publisher
 

Book/Create Book Model
Book model has been created.

Figure 4.8: Creating a book model.

Once at least a book model has been created as described above, books can be created ; an example is featured at Fig.4.9.

Books / Create :

Book/Create
Enter data on book:
Description
Model  Basic book model
 

Book/Create
Book has been created.

Figure 4.9: Creating a book.

This action also creates, in the current column two new headlines "Book - Main Book - Background" and "Book - Main Book - Foreground".

In this example, background and foreground headline models have been defined as follows :

[Headline models](#) / [Modify](#) / [Bk background](#) :

```
<style type="text/css">
  body {background-color:#AEABA0;}
</style>
<i>[This is Headline Model "Bk background"]</i>
```

[Headline models](#) / [Modify](#) / [Bk foreground](#) :

```
[Contents for Headline Model "Bk foreground"...]<br>
<b>[blah blah blah]</b>
```

The results of those simple book background and foreground models can be seen on the Headlines page :

[Headlines](#) / [Preview](#) :

[Book - Main Book - Background](#) :

[This is Headline Model "Bk background"]




[Book - Main Book - Foreground](#) :

[Contents for Headline Model "Bk foreground"...]
[blah blah blah]

4.3.3 Modifying a book

Just after it is created, a book only contains a background and foreground, as associated to its model.

Via [Modify](#), it becomes then possible to perform actions with sets of various icons (Fig.4.10). On this page, elements (including background and foreground) are called "menus".

-  : displays a preview of the element, just as the previews for Background and Foreground as displayed above. For "Background", it is the only icon available ;
-  ,  : allow to add a menu or submenu element before the previous element or after the next ;

Books / Modify / Main book :










Book/Modify			
Book: Main book			
Options	Menu	SubMenu	Menu Name
	0	0	Background
  	1	0	Foreground - (change name)
			

Figure 4.10: From here, book elements can be managed.

-  : allows to add a submenu under the menu (not available for submenus) ;
- ,  : with more than one menu (Background and Foreground excepted - icons not available for those) or submenu, allows to displace element before the previous one (if not the first in the list) or after the next one (if not the last in the list) ;
-  : removes any element that is not Background or Foreground.

4.4 Notice boards



User group rights for handling Notice Boards are detailed at 1.3.16.

Notice boards gather a collection of modifications that have been brought a site ; a notice board is associated with a column (1.4) and can be published on a headline as a list of links to every object that has been updated.

4.4.1 Creating and managing notice boards

An example creation of a notice board is featured at Fig.4.11 ; the maximum number of days modifications remain in the notice board is given. Then, it is assessed, amongst a list of headlines in all columns, a specific headline or none at all.

After their creation, notice boards can be modified or deleted thanks to the eponymous links.

4.4.2 Association with a column

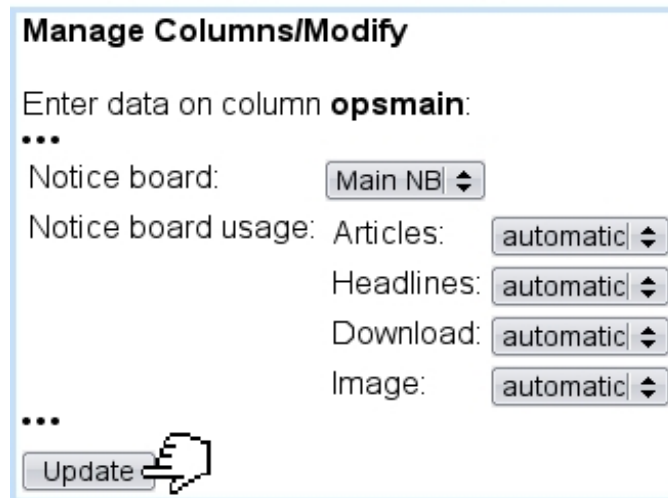
For a notice board to be used in the context of a column, the latter must be updated to specify it, in the following way :

[Notice board / Add entry](#) :



Figure 4.11: Creating a general notice board.

[Columns / Modify / opsmain](#) :



Menus there also allow to define individually, for articles, headlines, downloads and images, whether entries are done manually or automatically.

4.4.3 Notice board entries

If the notice board usage menus as seen in the previous images have been set 'automatic', this nonetheless does not prevent user from adding entries manually, as featured in Fig.4.12.

There, a table with all the latest modifications on the associated column's headline(s) is shown ; these are click-links, that, when clicked, add the modification to the notice

board, rendering it in normal font.

[Notice board / Add entry](#) :

Notice Board/Add
User: **Admin**, Column: **opsmain**

Select the entry you would like to add to the notice board:

Date	Action	Object type	Object Description
6/4/2011 14:15	Update	Headline	Subscribe 
6/4/2011 14:12	Create	Headline	Subscribe 
...	



Date	Action	Object type	Object Description
6/4/2011 14:15	Update	Headline	Subscribe
6/4/2011 14:12	Create	Headline	Subscribe
...	

Figure 4.12: Manually adding two entries to this notice board.

Entries can be removed from the notice board via [Delete entry](#).

4.5 Editions



User group rights for Edition handling are detailed at 1.3.10.



When creating a column (Fig.1.6), the Administrator can indicate whether the new column will be working in edition mode. To activate this mode for an already-existing column, s/he has to create a default edition for the column in the Blue Chameleon database.

The Edition mode is dedicated to publishers who wish to update their website and a periodical basis (e.g. a magazine that publishes every month an extract of their articles from the current paper edition).

In this mode, Blue Chameleon CMS supports different versions of a set of pages in the same column. These sets are stored in Editions (e.g. the index page leading to the article extracts will exist in editions January, February, etc.). When opening up a new

edition, all the headlines of the current column are copied to the new column.

Only documents from the active edition can be published. In order to replace the current website with the new edition, the new edition must be marked as 'active edition' and then be published.

For instance, let us suppose a column has been created with "work with editions" ticked with "April" as the name of the first edition ; with this column being set as the current one, the following can be seen :

[Edition](#) :

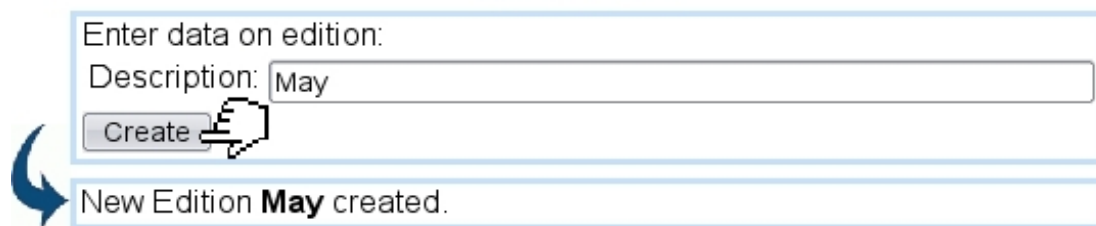
Edition: [Modify](#) - [New](#) - [Set active](#) - [Delete](#) (Current: April) (Active: April)


This current edition name, as well as any other, can afterwards be updated through [Edition](#) / [Modify](#).

4.5.1 General edition management

The process of creating a new edition is simple :

[Edition](#) / [New](#) :



Enter data on edition:
Description:
 

New Edition **May** created.

Once more than one edition exist in the system, any edition can be set to 'current' viz. 'active' via [Edition](#) / [Change](#) viz. [Set active](#).

The [Delete](#) link also allows to remove any edition that is not current or active.

On the *Headline Modify Page*, the following display will be shown :

[Headlines](#) / [Modify](#) :



Select a page:
..... **April** [May](#)

While the current edition is displayed in bold, with its headlines further down, any other edition (e.g. [May](#)) can be clicked to alter its contents.

4.5.2 Publishing editions

Through the [Edition](#) / [Publish](#) path, the following is available :

Edition Publish

Click on 'Edition Publish' to publish all objects that have been modified.

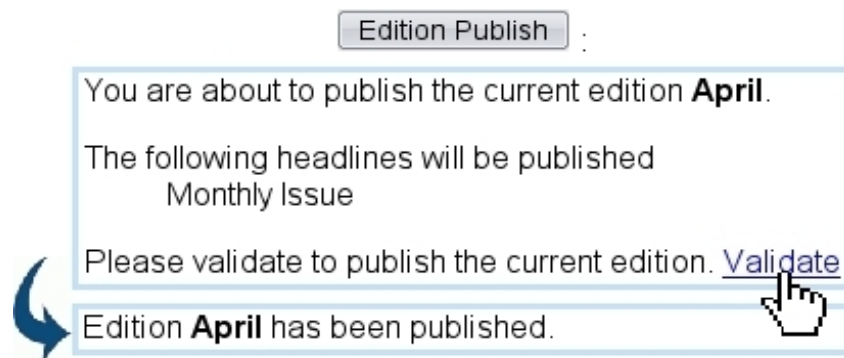
Detailed Publish

Click on 'Detailed Publish' to have the possibility to individually check the objects to be published.

There :

- clicking on the button will display a simple list of all objects that have been modified, with the only choice to validate their publication ;
- clicking on the button, on the other hand, will provide check-boxes to select headlines, articles,... and to check objects, exactly as in Fig.2.9.

The following shows for instance (as current and active editions are set to "April") the former process :



4.6 Options

This part of Blue Chameleon Content Management System, as accessed via [Options](#) is aimed at giving various information.

4.6.1 Version

This link allows to check the current versions of Blue Chameleon Content Management System and OSL, user's IP address as well as the versions (5.2.2) of all packages in use :

Options / Version :

Options/Version
Online Publishing System Version 2.27
Scripting Language Version 390
Your IP address is '10.1.1.37'.
Packages
Agenda version 1
Documents version 1
Festivals version 2
News version 1
Photo album version 2

4.6.2 System Logs



How actions are recorded and how long records are kept are defined at the Publisher settings (1.5).

Blue Chameleon Content Management System logs in the activity of its users. The system log contains information on the following actions :

- the modification, creation, deletion and publication of articles headlines, images and downloads ;
- the setting of ready-for-publication flag and the launching of scheduled publications ;
- the FTP deletion and FTP publication of articles, headlines, images and downloads ;
- the cancellation of FTP deletion and FTP publication ;
- the creation, modification, and deletion of users and user groups ;
- the users that have logged to Blue Chameleon.

As featured in Fig.4.13, this page provides several menus to narrow searches :

- user(s) of whom activity will be displayed : one's own log only, one's current group, logs of all groups and everything.



User group rights for displaying System Logs user-wise are detailed at 1.3.17.

- on which column the activity took place (or all columns) ;

- when this activity took place (the current day, or for the 2, 7 or 30 days) ;
- the kind of activity, related to content management or system (i.e. the managing of users, groups...) - or any.

Options / System Log :

Options/System Log

User **Column** **Period** **Type**
 User's own log ▾ All available columns ▾ Last 7 days ▾ Any type ▾

YourPublisher

Date	Column	User	Action	Object type	Object Description
4/4/2011 9:32	Publisher	Admin	Update	Group	Administrator
4/4/2011 9:01	opsmain	Admin	Logged in		
...					
1/4/2011 11:49	opsmain	Admin	Publish	Headline	Rock and Stuff
1/4/2011 11:49	opsmain	Admin	Publish	Headline	Login Page
1/4/2011 11:49	opsmain	Admin	Publish	Headline	Josh Jones Tulsa 2010
...					
1/4/2011 8:43	opsmain	Admin	Create	Headline	bk mod def headline
...					

Figure 4.13: Checking one's own activity for last week.

4.6.3 Expiration report

The eponymous link leads to a page where objects that have expired or are going to are listed ; by default, what decides if objects related to a column will expire is the setting as done when updating this column (1.4.3.2), which can then be fine-tuned for any headline on its Properties page (2.7.2).

4.7 Searching for content



User group rights for performing searches are detailed at 1.3.5.

Blue Chameleon Content Management System also offer a search tool that allows, as pictured in Fig.4.14, to perform searches in four different ways :

- by the name contents are published under ;

- by the publication folder (search then returns results by categories - headlines, images, etc.) ;
- by the object description ;
- by cumulating criteria object type, column and to/from dates (for which choice is creation, update and publish).

[Search / Search :](#)

Publisher - Search

Search by Publication Name
 Publication Name:

Search by Publication Path
 Publication Path:

Search by Object Description
 Object Description:

Search by Multiple Criteria
 Object type:
 Section:
 Date: from / / to / / Type

Search by Publication Name

Headlines:

Rubrique	Folder Name	Description	Last Publication	Last Modification
opsmain	Josh Jones Tulsa 2010	Josh Jones Tulsa 2010	1/4/2011	24/3/2011
Full-size pictures	Josh Tulsa 2010 FULL SIZE	Josh Tulsa 2010 FULL SIZE	4/4/2011	No date available

Figure 4.14: Performing a search on objects which publishing name begin by 'Josh'.

4.8 Management of objects

This page, as featured in Fig.4.15, offers the same tree-view structure and management functionalities than were already seen for images (3.1.3), downloads (3.4.2), headlines (2.7.3),... but for *any object*, as selected in the menu on the top.

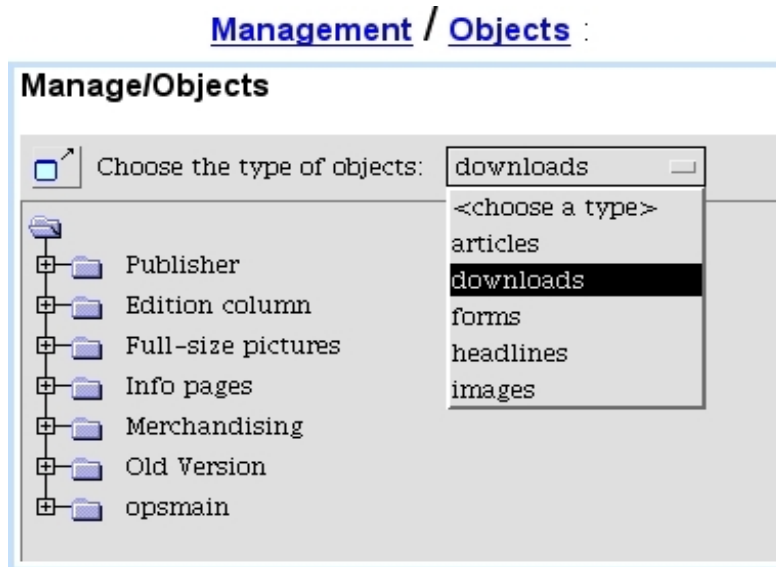


Figure 4.15: From here, any object can be managed.