

DocBookWiki



Contents

| | |
|---------------------------------------------------|---|
| | 2 |
| Package admin Procedural Elements | 2 |
| menu_items.php | 2 |
| user_data.php | 3 |
| Function get_user_data | 3 |
| Function save_user_data | 3 |
| Package admin Classes | 4 |
| Class admin | 4 |
| Method init | 4 |
| Method onParse | 4 |
| Class addbook | 4 |
| Method author_to_xml | 5 |
| Method get_initial_xml | 5 |
| Method get_legalnotice | 5 |
| Method keywords_to_xml | 5 |
| Method on_create | 5 |
| Class book_list | 6 |
| Method add_recordsets | 6 |
| Method init | 6 |
| Method onRender | 6 |
| Method on_delete | 6 |
| Method on_set_visible | 6 |
| Class import_doc | 6 |
| Method add_uploaded_files_rs | 7 |
| Method onRender | 7 |
| Method on_delete | 7 |
| Method on_import | 7 |
| Method on_refresh | 7 |
| Class books | 8 |
| Method add_book_list_rs | 8 |
| Method add_book_title | 8 |
| Method get_book_list_arr | 8 |
| Method init | 8 |
| Method onRender | 8 |
| Method on_select | 8 |
| Class uEdit | 8 |
| Method add_edit_rights | 8 |
| Method add_rs_langs | 8 |
| Method add_rs_nodes | 8 |
| Method add_user_data | 8 |
| Method get_langs | 9 |
| Method onRender | 9 |
| Method on_save | 9 |

| | |
|-----------------------------------------|----|
| Class uList | 9 |
| Method add_all_users_rs | 9 |
| Method add_users_rs | 9 |
| Method get_nr_of_recs | 9 |
| Method get_page_vars | 10 |
| Method init | 10 |
| Method onRender | 10 |
| Method on_add_user | 10 |
| Method on_delete | 10 |
| Method on_next | 10 |
| Method on_select | 10 |
| Method selectFirst | 10 |
| Class edit_menu | 10 |
| Method add_books_rs | 11 |
| Method add_subitems_rs | 11 |
| Method init | 11 |
| Method onRender | 11 |
| Method on_add_subitem | 11 |
| Method on_apply | 11 |
| Method on_cancel | 11 |
| Method on_delete | 11 |
| Method on_move_down | 12 |
| Method on_move_up | 12 |
| Method on_paste | 12 |
| Method on_select | 12 |
| Method on_update | 12 |
| Method select_item | 13 |
| Method transform | 13 |
| Method transform_menu | 13 |
| userEdit.php | 14 |
| Class userEdit | 14 |
| Var \$user_data | 14 |
| Method init | 14 |
| Method onParse | 14 |
| Method onRender | 14 |
| Method on_add | 15 |
| Method on_delete | 15 |
| Method on_save | 15 |
| Class userFilter | 15 |
| Method buildFilter | 16 |
| Method init | 16 |
| Method onParse | 16 |
| Class userList | 16 |
| Method add_users_rs | 16 |
| Method get_nr_of_recs | 16 |
| Method get_page_vars | 16 |
| Method init | 16 |
| Method onRender | 16 |
| Method on_add_new | 16 |

| | |
|---------------------------------------------------------|----|
| Method on delete | 16 |
| Method on next | 16 |
| Method on refresh | 17 |
| Method on select | 17 |
| Method selectFirst | 17 |
| modified_nodes.php | 18 |
| user_data.php | 19 |
| Class edit_rights | 19 |
| Method init | 19 |
| Method onRender | 19 |
| Class modified_nodes | 19 |
| Method add_has_accr | 20 |
| Method add_is_admin | 20 |
| Method get_arr_books_accr | 20 |
| Method get_arr_books_admin | 20 |
| Method get_title | 20 |
| Method has_approve_right | 21 |
| Method lang_match | 21 |
| Method node_match | 21 |
| Method onRender | 22 |
| Class user_data | 22 |
| Method add_rs_books_admin | 22 |
| Method onParse | 22 |
| Method onRender | 22 |
| Method on_change_passwd | 22 |
| Method on_save | 22 |
| Package DocBookWiki Procedural Elements | 24 |
| const.Debug.php | 24 |
| Define DEBUG_GOTO | 24 |
| Define DEBUG_INCLUDES | 24 |
| Define DEBUG_RECORDSETS | 24 |
| Define DEBUG_STATEVARS | 24 |
| Define DEBUG_TEMPLATES | 24 |
| Define DEBUG_WEBOBJECTS | 25 |
| Define EXECUTION_TIME_INFO | 25 |
| Define SHOW_ERROR_MESSAGES | 25 |
| Define SHOW_EXTERNAL_LINK | 25 |
| const.Options.php | 26 |
| Define \$const | 26 |
| Define APP_CONFIG_FILE | 26 |
| Define DB_TYPE | 26 |
| Define DISPLAY_CGI_VARS | 26 |
| Define FIRSTPAGE | 26 |
| Define NULL_VALUE | 26 |
| Define USES_DB | 26 |
| Define VAR_NOT_FOUND | 26 |
| const.Paths.php | 28 |
| Define ADMIN | 28 |
| Define BOOKS | 28 |

| | |
|----------------------------------------------------|----|
| Define CACHE | 28 |
| Define CONTENT | 28 |
| Define CSS | 28 |
| Define DOCBOOK | 28 |
| Define GRAPHICS | 28 |
| Define MENU | 28 |
| Define SCRIPTS | 28 |
| Define SVN | 28 |
| Define TPL | 28 |
| Define WEBAPP_PATH | 28 |
| Define WOBJ | 28 |
| Define WORKSPACE | 28 |
| Define WS_BOOKS | 28 |
| Define WS_CACHE | 28 |
| Define XSLT | 28 |
| func.languages.php | 29 |
| Function get_arr_languages | 29 |
| languages.php | 30 |
| import_doc.php | 31 |
| Function upload_files | 31 |
| Package DocBookWiki Classes | 32 |
| Class books | 32 |
| Method onRender | 32 |
| Class help | 32 |
| Method afterParse | 32 |
| Method get_book_title | 32 |
| Method get_langs | 33 |
| Method on_select | 33 |
| Class languages | 33 |
| Method init | 33 |
| Method onParse | 33 |
| Method onRender | 33 |
| Method on_select | 33 |
| Class main | 34 |
| Method check_book_id | 34 |
| Method get_book_title | 34 |
| Method get_langs | 34 |
| Method init | 35 |
| Method onRender | 35 |
| Method on_approve | 35 |
| Method on_book | 35 |
| Method on_docbook | 35 |
| Method on_search | 35 |
| Method on_xref | 35 |
| Class search | 35 |
| Method get_book_list | 36 |
| Method get_nr_of_recs | 36 |
| Method get_page_vars | 36 |
| Method get_query | 36 |

| | |
|------------------------------------------------------------|----|
| Method get_results_rs | 36 |
| Method init | 36 |
| Method onParse | 36 |
| Method onRender | 36 |
| Method on_help | 36 |
| Method on_new_search | 36 |
| Method on_next | 36 |
| | 38 |
| Package docbook Procedural Elements | 38 |
| docbook.php | 38 |
| funcs.php | 39 |
| Function book_fixed_to_tag | 39 |
| Function file_content_html | 39 |
| Function file_content_xml | 39 |
| Function get_arr_navigation | 40 |
| Function get_date_str | 40 |
| Function get_node_state | 40 |
| Function get_state_filename | 40 |
| Function is_locked | 40 |
| Function locked_by_somebody | 41 |
| Function process_index | 41 |
| Function set_node_lock | 41 |
| Function set_node_status | 42 |
| Function update_cache | 42 |
| Function write_node_state | 42 |
| langs.php | 43 |
| Package docbook Classes | 44 |
| Class docbook | 44 |
| Method add_accessright_vars | 44 |
| Method get_button_vars | 44 |
| Method get_edit_rights | 44 |
| Method init | 44 |
| Method is_admin | 44 |
| Method lang_match | 45 |
| Method node_match | 45 |
| Method onParse | 45 |
| Method onRender | 45 |
| Method on_set lng | 45 |
| Method on_set mode | 45 |
| Method on_set node | 45 |
| Method on_set node id | 46 |
| Class langs | 46 |
| Method onRender | 46 |
| Class view | 46 |
| Method init | 46 |
| Method onParse | 46 |
| Method setup_webnotes | 46 |
| func.popup_window.php | 47 |

| | |
|------------------------------------------------|----|
| tags.php | 48 |
| Class admin | 48 |
| Method add_rs_modified_nodes | 48 |
| Method get_title | 48 |
| Method onRender | 48 |
| Method on_update_booklist | 49 |
| Method on_update_downloadables | 49 |
| Method on_update_search_index | 49 |
| Class tags | 49 |
| Method add_rs_tags | 49 |
| Method get_book_dir | 49 |
| Method get_current_tag | 49 |
| Method get_head_revision | 49 |
| Method get_revision_date | 50 |
| Method get_revision_nr | 50 |
| Method get_svn_url | 50 |
| Method is_fixed | 50 |
| Method onRender | 50 |
| Method on_add_tag | 50 |
| Method on_del_tag | 50 |
| Method on_set_date | 50 |
| Method on_set_rev | 50 |
| Method on_set_tag | 50 |
| Method set_revision | 50 |
| Method write_fixed_file | 51 |
| approve.php | 52 |
| Class approve | 52 |
| Method get_content_diff | 52 |
| Method get_svn_status | 52 |
| Method msg_book_fixed_to_tag | 52 |
| Method non_recursive_commit | 53 |
| Method non_recursive_revert | 53 |
| Method onRender | 53 |
| Method on_commit | 53 |
| Method on_revert | 53 |
| Method on_set_lock | 53 |
| Method recursive_commit | 53 |
| Method recursive_revert | 53 |
| Method set_all_synchronized | 53 |
| Class revisions | 54 |
| Method add_file_revision | 54 |
| Method add_rev_diff | 54 |
| Method add_rs_revs | 54 |
| Method add_rs_tags | 54 |
| Method clear | 54 |
| Method get_revision | 54 |
| Method get_svn_url | 54 |
| Method init | 54 |
| Method onRender | 54 |

| | |
|---------------------------------------------------------|----|
| Method on hide | 54 |
| Method on save | 55 |
| Method on set date | 55 |
| Method on set rev | 55 |
| Method on set tag | 55 |
| Method on show | 55 |
| convert from xml.php | 56 |
| Function xml to html | 56 |
| Function xml to text | 56 |
| convert to xml.php | 57 |
| Function get id title | 57 |
| Function html to xml | 57 |
| Function is wellformed | 57 |
| Function latex to xml | 57 |
| Function texi to xml | 57 |
| Function text to xml | 57 |
| Function xml to xml | 58 |
| edit content.php | 59 |
| func.popup window.php | 60 |
| Function popup window | 60 |
| process content.php | 61 |
| Function compact cdata | 61 |
| Function expand cdata | 61 |
| Function get node content | 61 |
| Function putback cdata | 62 |
| Function putback comments | 62 |
| Function strip cdata | 62 |
| Function strip comments | 62 |
| validate.php | 64 |
| Function display error messages | 64 |
| Function get short xml line | 64 |
| Function validate xml | 64 |
| edit.php | 65 |
| edit funcs.php | 66 |
| Function add subsections rs | 66 |
| Function add to modified nodes | 66 |
| Function changeid movefolders | 66 |
| Function changeid setid | 66 |
| Function changeid updatecache | 67 |
| Function change id validate | 67 |
| Function create new node | 67 |
| Function delete removefolders | 67 |
| Function delete removefolders all langs | 67 |
| Function get node references | 68 |
| Function lock index | 68 |
| Function process index node | 68 |
| Function remove from modified nodes | 68 |
| Function transform index | 68 |
| Function unlock index | 69 |

| | |
|---------------------------------------------------------|----|
| Function updatenavig_movedown | 69 |
| Function updatenavig_moveup | 69 |
| Function update_cache_files | 69 |
| Function update_cache_files_all_langs | 69 |
| Function update_navigation | 69 |
| Function update_navigation_all_langs | 69 |
| Function update_subnodes_html | 70 |
| Function update_subnodes_html_all_langs | 70 |
| Class edit | 70 |
| Method add_nodetypes_rs | 71 |
| Method onParse | 71 |
| Method onRender | 71 |
| Method on_add_subnode | 71 |
| Method on_change_id | 71 |
| Method on_change_title | 71 |
| Method on_delete | 71 |
| Method on_move_down | 71 |
| Method on_move_up | 71 |
| Method on_set_lock | 72 |
| Class edit_content | 72 |
| Method add_tab_items | 72 |
| Method error_on_save | 72 |
| Method init | 72 |
| Method onRender | 72 |
| Method on_save | 72 |
| Method on_set_mode | 72 |
| Class edit_media | 73 |
| Method get_media_path | 73 |
| Method onRender | 73 |
| Method on_delete | 73 |
| Method on_rename | 73 |
| Method on_upload | 73 |
| package.WikiConverter.php | 74 |
| Class Block | 74 |
| Var \$block_title | 75 |
| Var \$block_type | 75 |
| Constructor Block | 75 |
| Method line_to_html | 75 |
| Method line_to_xml | 75 |
| Method to_html | 75 |
| Method to_xml | 75 |
| Class Info | 76 |
| Var \$abstract | 76 |
| Var \$authors | 76 |
| Var \$copyright | 76 |
| Var \$keywords | 76 |
| Constructor Info | 76 |
| Method parse | 76 |
| Method parse_abstract | 76 |

| | |
|-----------------------------------------|----|
| Method parse author | 77 |
| Method parse copyright | 77 |
| Method parse dummy | 77 |
| Method parse keywords | 77 |
| Method render abstract | 77 |
| Method render authors | 77 |
| Method render copyright | 77 |
| Method render keywords | 77 |
| Method to xml | 77 |
| Class Listing | 77 |
| Var \$bullet | 78 |
| Var \$indent | 78 |
| Constructor Listing | 78 |
| Method to html | 78 |
| Method to xml | 78 |
| Class ListItem | 78 |
| Constructor ListItem | 79 |
| Method to html | 79 |
| Method to xml | 79 |
| Class Paragraph | 79 |
| Constructor Paragraph | 79 |
| Method to html | 79 |
| Method to xml | 79 |
| Class Text | 80 |
| Constructor Text | 80 |
| Method line to html | 80 |
| Method line to xml | 80 |
| Class Tpl | 80 |
| Var \$contents | 81 |
| Var \$id | 81 |
| Var \$type | 81 |
| Constructor Tpl | 81 |
| Method get_subtemplates | 81 |
| Method get_tpl_id | 81 |
| Method line to html | 81 |
| Method line to xml | 82 |
| Method to html | 82 |
| Method to html table | 82 |
| Method to xml | 82 |
| Class WikiParser | 82 |
| Var \$arr_lines | 82 |
| Var \$idx | 83 |
| Constructor WikiParser | 83 |
| Method append_line | 83 |
| Method block_end | 83 |
| Method block_start | 83 |
| Method empty_line | 83 |
| Method end_of_lines | 83 |
| Method get_bullet | 83 |

| | |
|---------------------------------------------------------------|----|
| Method get_current_line | 83 |
| Method get_indentation | 83 |
| Method get_listitem_mark | 84 |
| Method link_template | 84 |
| Method listitem_end | 84 |
| Method listitem_start | 84 |
| Method list_end | 84 |
| Method list_start | 84 |
| Method next_line | 84 |
| Method nonempty_line | 84 |
| Method parse_block | 85 |
| Method parse_file | 85 |
| Method parse_info | 85 |
| Method parse_lines | 85 |
| Method parse_list | 85 |
| Method parse_listitem | 85 |
| Method parse_paragraph | 85 |
| Method parse_string | 85 |
| Method parse_text | 85 |
| Method template_list | 85 |
| Method to_tree | 85 |
| Method tpl_to_tree | 86 |
| fun.regex_replace.php | 87 |
| Function convert_menu_items | 87 |
| Function mediaobject | 87 |
| Function regex_replace | 87 |
| Package webobjects Procedural Elements | 90 |
| tabs1_items.php | 90 |
| Package webobjects Classes | 91 |
| Class tabs1 | 91 |
| Method add_items_list | 91 |
| Method init | 91 |
| Method onRender | 91 |
| Method on_select | 91 |
| Appendices | 92 |
| Appendix A - Class Trees | 93 |
| DocBookWiki | 93 |
| docbook | 94 |
| admin | 96 |
| webobjects | 98 |

Package admin

```
<include src="{{.}}header.html" />
```

```
<webbox id="admin">  
  <include src="{{.}}file" />  
</webbox>
```

```
<include src="{{TPL}}main/footer.html" />
```

Package admin Procedural Elements

menu_items.php

The `$menu_items` array contains the items of the tabs1.

- **Package** admin

user_data.php

- **Package** admin

function get_user_data(\$username) [*line 29*]

Function Parameters:

- **\$username**

function save_user_data(\$user_data) [*line 44*]

Function Parameters:

- **\$user_data**

Package admin Classes

Class admin

[line 31]

Modules and interfaces that are used by the superuser and by the book admins. Is accessed through 'admin.php'.

- **Package** admin

void function admin::init() *[line 33]*

void function admin::onParse() *[line 47]*

Class addbook

[line 35]

This module is used to import a new book/article in the system or to create a new one. A list of the current books in the system is displayed as well and it is also possible to remove some books from the system. Only the superuser can access this module.

- **Package** admin
- **Sub-Package** addbook

void function addbook::author_to_xml(\$str) [line 136]

Function Parameters:

- **\$str**

Gets a string like this: 'Surname, Firstname, email'

and returns an xml like this: <author> <firstname>Firstname</firstname>
<surname>Surname</surname> <affiliation>
<address><email>email</email></address> </affiliation>
</author>

void function addbook::get_initial_xml(\$event_args) [line 61]

Function Parameters:

- **\$event_args**

constructs and returns an intial xml document

void function addbook::get_legalnotice(\$author) [line 187]

Function Parameters:

- **\$author**

void function addbook::keywords_to_xml(\$str) [line 172]

Function Parameters:

- **\$str**

Gets a comma separated list of keywords: 'k1, k2, k3,'

and returns an xml like this: <keywordset> <keyword>k1</keyword>
<keyword>k2</keyword> <keyword>k3</keyword> </keywordset>

void function addbook::on_create(\$event_args) [line 37]

Function Parameters:

- **\$event_args**

Class book_list

[line 31]

Displays a list of the current books in the system.

- **Package** admin
- **Sub-Package** addbook

void function book_list::add_recordsets() [line 58]

void function book_list::init() [line 33]

void function book_list::onRender() [line 53]

void function book_list::on_delete(\$event_args) [line 44]

Function Parameters:

- **\$event_args**

void function book_list::on_set_visible(\$event_args) [line 38]

Function Parameters:

- **\$event_args**

Class import_doc

[line 31]

Import a new book or article in the system.

- **Package** admin
- **Sub-Package** addbook

void function import_doc::add_uploaded_files_rs() [line 93]
create and add the recordset 'uploaded_files'

void function import_doc::onRender() [line 87]
void function import_doc::on_delete(\$event_args) [line 40]
Function Parameters:

- **\$event_args**

delete an uploaded file

void function import_doc::on_import(\$event_args) [line 49]
Function Parameters:

- **\$event_args**

void function import_doc::on_refresh(\$event_args) [line 34]
Function Parameters:

- **\$event_args**

refresh the list after uploading a file

Class books

[line 30]

The books that the user can administrate.

- **Package** admin
- **Sub-Package** books

void function books::add_book_list_rs() *[line 72]*

void function books::add_book_title() *[line 50]*

void function books::get_book_list_arr() *[line 94]*

void function books::init() *[line 32]*

void function books::onRender() *[line 44]*

void function books::on_select(\$event_args) *[line 38]*

Function Parameters:

- **\$event_args**

Class uEdit

[line 30]

Admin the edit rights of a user(editor) of a book.

- **Package** admin
- **Sub-Package** books

void function uEdit::add_edit_rights() *[line 65]*

void function uEdit::add_rs_langs() *[line 85]*

void function uEdit::add_rs_nodes() *[line 102]*

void function uEdit::add_user_data() *[line 56]*

get the user data and add them to the template

void function uEdit::get_langs(\$book_id) [line 129]

Function Parameters:

- **\$book_id**

Returns an array of languages (en,fr,it,al) in which the book is available.

void function uEdit::onRender() [line 47]

void function uEdit::on_save(\$event_args) [line 33]

Function Parameters:

- **\$event_args**

save the changes

Class uList

[line 30]

The list of the users(editors) that can edit a book.

- **Package** admin
- **Sub-Package** books

void function uList::add_all_users_rs() [line 121]

Constructs and adds to the webpage the recordset 'all_users' with all the users of the site.

void function uList::add_users_rs() [line 142]

Constructs and adds to the webpage the recordset 'users'.

void function uList::get_nr_of_recs() [line 177]

void function uList::get_page_vars() [*line 185*]
void function uList::init() [*line 32*]
void function uList::onRender() [*line 112*]
void function uList::on_add_user(\$event_args) [*line 84*]

Function Parameters:

- **\$event_args**

void function uList::on_delete(\$event_args) [*line 69*]

Function Parameters:

- **\$event_args**

void function uList::on_next(\$event_args) [*line 56*]

Function Parameters:

- **\$event_args**

void function uList::on_select(\$event_args) [*line 63*]

Function Parameters:

- **\$event_args**

void function uList::selectFirst() [*line 43*]

set currentUser as the first user in the list

Class edit_menu

[*line 32*]

Modify the menu of the categories and books.

Only the superuser can access this module.

- **Package** admin
- **Sub-Package** edit_menu

void function edit_menu::add_books_rs() [line 233]

Add a recordset of all the books, which will be displayed in a listbox.

void function edit_menu::add_subitems_rs() [line 210]

add a recordset of the subitems of the selected item

void function edit_menu::init() [line 34]

void function edit_menu::onRender() [line 203]

void function edit_menu::on_add_subitem(\$event_args) [line 135]

Function Parameters:

- **\$event_args**

add a new subitem to the current item

void function edit_menu::on_apply(\$event_args) [line 173]

Function Parameters:

- **\$event_args**

apply the modifications to the main menu

void function edit_menu::on_cancel(\$event_args) [line 193]

Function Parameters:

- **\$event_args**

discard any modifications and get a copy of the main menu

void function edit_menu::on_delete(\$event_args) [line 112]

Function Parameters:

- **\$event_args**

delete the current item and set the parent item as the current one

void function edit_menu::on_move_down(\$event_args) [line 128]

Function Parameters:

- **\$event_args**

move down the given item

void function edit_menu::on_move_up(\$event_args) [line 121]

Function Parameters:

- **\$event_args**

move up the given item

void function edit_menu::on_paste(\$event_args) [line 143]

Function Parameters:

- **\$event_args**

add any items in clipboard as subitems of the current item

void function edit_menu::on_select(\$event_args) [line 46]

Function Parameters:

- **\$event_args**

select an item for editing

void function edit_menu::on_update(\$event_args) [line 102]

Function Parameters:

- **\$event_args**

save modifications in id and caption of the current item

void function edit_menu::select_item(\$item_id) [line 56]

Function Parameters:

- **\$item_id**

Get the 'bookid' and the 'caption' of the selected item and refresh the state variables.

void function edit_menu::transform(\$transformer, [\$arr_params = array()]) [line 82]

Function Parameters:

- **\$transformer**
- **\$arr_params**

Applies the given xsl transformer to menu.xml and returns the result. \$arr_params is an associative array of parameters.

void function edit_menu::transform_menu(\$transformer, [\$arr_params = array()]) [line 68]

Function Parameters:

- **\$transformer**
- **\$arr_params**

userEdit.php

- **Package** admin
- **Sub-Package** users

include_once **FORM_PATH.**'formWebObj.php' *[line 26]*

include_once [user_data.php](#)*[line 25]*

Class userEdit

[line 38]

userEdit modifies the details of a user (username, password, name, email and which books can admin).

Users are those that can login at edit.php and admin.php interfaces (they could have been called also editors). Only the superuser can access the users module.

- **Package** admin
- **Sub-Package** users

userEdit::\$user_data

```
mixed = array(  
    'username' => "",  
    'password' => "",  
    'name' => "",  
    'email' => "",  
    'books' => ""  
)[line 40]
```

void function userEdit::init() *[line 48]*

void function userEdit::onParse() *[line 146]*

void function userEdit::onRender() *[line 163]*

void function userEdit::on_add(\$event_args) [*line 54*]

Function Parameters:

- **\$event_args**

add the new user in database

void function userEdit::on_delete(\$event_args) [*line 97*]

Function Parameters:

- **\$event_args**

delete the current user

void function userEdit::on_save(\$event_args) [*line 114*]

Function Parameters:

- **\$event_args**

save the changes

Class userFilter

[*line 34*]

userFilter is used to select the users that are displayed in the list.

Users are those that can login at edit.php and admin.php interfaces (they could have been called also editors). Only the superuser can access the users module.

- **Package** admin
- **Sub-Package** users

void function userFilter::buildFilter() [line 55]
void function userFilter::init() [line 36]
void function userFilter::onParse() [line 50]

Class userList

[line 34]

userList displays the list of system users.

Users are those that can login at edit.php and admin.php interfaces (they could have been called also editors). Only the superuser can access the users module.

- **Package** admin
- **Sub-Package** users

void function userList::add_users_rs() [line 109]

Constructs and adds to the webpage the recordset 'users'.

void function userList::get_nr_of_recs() [line 144]
void function userList::get_page_vars() [line 152]
void function userList::init() [line 36]
void function userList::onRender() [line 102]
void function userList::on_add_new(\$event_args) [line 96]

Function Parameters:

- **\$event_args**

void function userList::on_delete(\$event_args) [line 82]

Function Parameters:

- **\$event_args**

void function userList::on_next(\$event_args) [line 60]

Function Parameters:

- **\$event_args**

void function userList::on_refresh(\$event_args) [*line 67*]

Function Parameters:

- **\$event_args**

void function userList::on_select(\$event_args) [*line 76*]

Function Parameters:

- **\$event_args**

void function userList::selectFirst() [*line 47*]

set currentUser as the first user in the list

modified_nodes.php

- **Package** admin
- **Sub-Package** user_data

include_once [user_data.php](#)*[line 25]*

user_data.php

- **Package** admin
- **Sub-Package** user_data

include_once [SCRIPTS.'user_data.php'](#) *[line 25]*

Class edit_rights

[line 31]

Displays the edit rights of the user.

- **Package** admin
- **Sub-Package** user_data

void function edit_rights::init() *[line 33]*

void function edit_rights::onRender() *[line 38]*

Class modified_nodes

[line 33]

Displays a list of the modified nodes that can be approved by the user.

- **Package** admin
- **Sub-Package** user_data

void function modified_nodes::add_has_accr(&\$rs, \$arr_books_accr) [line 138]

Function Parameters:

- **&\$rs**
- **\$arr_books_accr**

Find the modified nodes that can be approved by user and add them in the recordset.

void function modified_nodes::add_is_admin(&\$rs, \$arr_books_admin) [line 81]

Function Parameters:

- **&\$rs**
- **\$arr_books_admin**

Add in the recordset all the modified nodes in the books for which the user is admin.

void function modified_nodes::get_arr_books_accr(\$arr_books_admin) [line 111]

Function Parameters:

- **\$arr_books_admin**

Return an array with the books where the user has some access rights and is not admin.

void function modified_nodes::get_arr_books_admin() [line 65]

return an array of the books for which the user is admin

void function modified_nodes::get_title(\$book_id, \$lng, \$node_path) [line 239]

Function Parameters:

- **\$book_id**
- **\$lng**

- **\$node_path**

void function modified_nodes::has_approve_right(\$lng, \$node_path, \$arr_access_rights) [line 179]

Function Parameters:

- **\$lng**
- **\$node_path**
- **\$arr_access_rights**

Returns true if the user has approve right on the given node.

void function modified_nodes::lang_match(\$lng, \$lang_list) [line 229]

Function Parameters:

- **\$lng**
- **\$lang_list**

Returns true if one of the language ids in the list matches the given language, otherwise returns false. The lang_list is a comma separated list of languages, or 'ALL'.

'ALL' matches any language.

void function modified_nodes::node_match(\$node_path, \$node_list) [line 211]

Function Parameters:

- **\$node_path**
- **\$node_list**

Returns true if one of the node_path expressions in the list matches the given node path, otherwise returns false.

The node_list is a comma separated list of node_path expressions (which are actually regular expressions), or is the string 'ALL'. 'ALL' matches any node path.

void function modified_nodes::onRender() [line 41]

First find the books where the user has some access rights.

Then find the modified nodes in each of these books. Next, check whether the user has approve rights on these modified nodes, and if yes, add them to the recordset 'modified_nodes'.

Class user_data

[line 35]

The user can change his data: name, email and password; he can also see which books he can manage, his edit rights and a list of nodes that are modified and he can approve.

- **Package** admin
- **Sub-Package** user_data

void function user_data::add_rs_books_admin() [line 74]

Add to webPage a recordset of the books for which the user is admin.

void function user_data::onParse() [line 57]

void function user_data::onRender() [line 62]

void function user_data::on_change_passwd(\$event_args) [line 45]

Function Parameters:

- **\$event_args**

void function user_data::on_save(\$event_args) [line 37]

Function Parameters:

- **\$event_args**

Package DocBookWiki Procedural Elements

const.Debug.php

The following constants are debug constants.

- **Package** DocBookWiki

DEBUG_GOTO = false *[line 10]*

if this constant is true, the framework displays an alert
(each time that the function GoTo() is called)

DEBUG_INCLUDES = false *[line 28]*

if this constant is true, the framework outputs a box
around each file that is included in the page

DEBUG_RECORDSETS = false *[line 40]*

if this constant is true, the framework outputs all the
recordsets of the page and their contents

DEBUG_STATEVARS = false *[line 16]*

if this constant is true, the framework outputs the state
variables of each webbox in HTML format

DEBUG_TEMPLATES = false *[line 22]*

if this constant is true, the framework outputs the tree
structure of the templates of the page

DEBUG_WEBOBJECTS = false *[line 34]*

if this constant is true, the framework outputs a box
around each webobject that is used in the page

EXECUTION_TIME_INFO = false *[line 46]*

if this constant is true, the framework outputs information
about the execution time of several processes

SHOW_ERROR_MESSAGES = true *[line 64]*

if this constant is false than the framework suppresses

all the error and warning messages (that are generated by the framework); usually this is true during the development and is switched to false when the application is deployed, in order not to confuse the users of the application with error messages that have no meaning for them

SHOW_EXTERNAL_LINK = false *[line 54]*

If this constant is true, the framework outputs the external link that can be used to jump right to the current page of the application from any external page.

See also DISPLAY_CGI_VARS at const.Options.php.

const.Options.php

The constants defined in this file change the behaviour of the framework. You can change the values of the constants according to the instructions given in comments.

- **Package** DocBookWiki

`$const = $val` *[line 68]*

`APP_CONFIG_FILE = "books.conf"` *[line 54]*

The file that contains additional configuration options, which are specific for the application.

`DB_TYPE = "MySQL"` *[line 26]*

this constant defines the type of DB that the application uses it can be:
"MySQL", "Oracle", "ODBC",
"ProgreSQL", etc.
(except "MySQL", the others are not implemented yet)

`DISPLAY_CGI_VARS = false` *[line 48]*

When this constant is true, then the CGI vars are displayed at the URL window of the browser. See also SHOW_EXTERNAL_LINK at const.Debug.php.

`FIRSTPAGE = "main/main.html"` *[line 12]*

This is the first page of the application. The framework looks for it at the template folder (specified by TPL).

`NULL_VALUE = "NULL"` *[line 33]*

This constant is the value returned by the framework for a DB variable that has a NULL value. It can be "", "NULL", NULL, etc.

`USES_DB = false` *[line 19]*

if this constant is true, the framework will load the DB component and will open a default connection with the db specified in the file 'config/const.DB.php'

`VAR_NOT_FOUND = "{var_name}"` *[line 41]*

This constant sets the format of the error message that is displayed when a `{{variable}}` is not found. 'var_name' is replaced by the actual variable name. Examples: `"var_name' is not defined"`, `""`, `"undefined"`, etc. It cannot contain `"{{var_name}}"` inside.

const.Paths.php

- **Package** DocBookWiki

```
ADMIN = TPL.'admin/' [line 9]
BOOKS = CONTENT.'books/xml/' [line 16]
CACHE = CONTENT.'books/cache/' [line 18]
CONTENT = 'content/' [line 15]
CSS = 'css/' [line 5]
DOCBOOK = TPL.'docbook/' [line 8]
GRAPHICS = 'graphics/' [line 4]
MENU = TPL.'main/menu/' [line 7]
SCRIPTS = TPL.'scripts/' [line 10]
SVN = CONTENT.'books/svn/' [line 17]
TPL = 'templates/' [line 6]
WEBAPP_PATH = 'web_app/' [line 3]
W OBJ = 'webojects/' [line 13]
WORKSPACE = 'content/workspace/' [line 20]
WS_BOOKS = WORKSPACE.'xml/' [line 21]
WS_CACHE = WORKSPACE.'cache/' [line 22]
XSLT = 'xsl_transform/' [line 12]
```

func.languages.php

- **Package** DocBookWiki

function get_arr_languages() *[line 33]*

This function returns an associated array of language codes and language details (like language name, codeset, etc.).

It can be used like this: `$langs = get_arr_languages(); $name = $langs['en']['name']; $codeset = $langs['en']['codeset'];`

languages.php

- **Package** DocBookWiki

`include_once TPL.'languages/func.languages.php'` [*line 25*]

import_doc.php

- **Package** DocBookWiki

function upload_files() [*line 31*]

Upload the files 'xml_file' and 'media_files'.

Sets the global variables \$fnames (files that are being uploaded) and \$upload_msg (successfullnes of upload), which are used to give a feedback message to the user.

Package DocBookWiki Classes

Class books

[line 28]

A listbox of the books/articles in the system.

- **Package** DocBookWiki

void function books::onRender() *[line 30]*

Class help

[line 28]

Displays a section of docbookwiki_guide.

- **Package** DocBookWiki

void function help::afterParse() *[line 61]*

void function help::get_book_title(\$book_id, \$lng) *[line 85]*

Function Parameters:

- **\$book_id**

- **\$lng**

void function help::get_langs(\$book_id) [line 74]

Function Parameters:

- **\$book_id**

Returns an array of languages (en,fr,it,al) in which the book is available.

void function help::on_select(\$event_args) [line 31]

Function Parameters:

- **\$event_args**

this event is called by xref.php

Class languages

[line 30]

Languages of the application interface.

- **Package DocBookWiki**

void function languages::init() [line 32]

void function languages::onParse() [line 45]

void function languages::onRender() [line 53]

void function languages::on_select(\$event_args) [line 38]

Function Parameters:

- **\$event_args**

Class main

[line 28]

The starting point for constructing the web page.

- **Package** DocBookWiki

void function main::check_book_id(\$book_id) [line 60]

Function Parameters:

- **\$book_id**

Checks whether the given book_id exists or not.

If it does not exist, display an error message and return a book_id that does exist.

void function main::get_book_title(\$book_id, \$lng) [line 251]

Function Parameters:

- **\$book_id**
- **\$lng**

void function main::get_langs(\$book_id) [line 240]

Function Parameters:

- **\$book_id**

Returns an array of languages (en,fr,it,al) in which the book is available.

void function main::init() [line 30]
void function main::onRender() [line 231]
void function main::on_approve(\$event_args) [line 100]
Function Parameters:

- **\$event_args**

void function main::on_book(\$event_args) [line 72]
Function Parameters:

- **\$event_args**

void function main::on_docbook(\$event_args) [line 138]
Function Parameters:

- **\$event_args**

void function main::on_search(\$event_args) [line 122]
Function Parameters:

- **\$event_args**

void function main::on_xref(\$event_args) [line 183]
Function Parameters:

- **\$event_args**

this event is called by xref.php

Class search

[line 28]

Search in the book(s) and display the results.

- **Package** DocBookWiki

void function search::get_book_list() [*line 128*]

Read the file books/book_list and return the associative array \$arr_books, with keys 'book_id:lng' and with values 'book_title'.

void function search::get_nr_of_recs() [*line 170*]

void function search::get_page_vars() [*line 193*]

void function search::get_query() [*line 145*]

void function search::get_results_rs() [*line 80*]

Returns a recordset of search results.

void function search::init() [*line 30*]

void function search::onParse() [*line 58*]

void function search::onRender() [*line 67*]

void function search::on_help(\$event_args) [*line 47*]

Function Parameters:

- **\$event_args**

void function search::on_new_search(\$event_args) [*line 41*]

Function Parameters:

- **\$event_args**

void function search::on_next(\$event_args) [*line 52*]

Function Parameters:

- **\$event_args**

Package docbook

```
<div class="docbook">  
  <include src="{{.}}header.html" />  
  <hr />  
  <include src="{{.}}{{node}}" />  
  <hr />  
  <include src="{{.}}footer.html" />  
</div>  
</w
```

Package docbook Procedural Elements

docbook.php

- **Package** docbook

```
include_once __FILE__ [line 25]
```

funcs.php

This file has some common functions.

- **Package** docbook

function book_fixed_to_tag(\$book_id, \$lng) *[line 372]*

Function Parameters:

- **\$book_id**
- **\$lng**

Check whether the public copy of the book is fixed to a certain tag or revision and return it. If book is not fixed, return false.

function file_content_html([\$cache_path = WS_CACHE], [\$node_path = UNDEFINED], [\$lng = UNDEFINED]) *[line 61]*

Function Parameters:

- **\$cache_path**
- **\$node_path**
- **\$lng**

Returns the path of the content_lng.html for the given node and language.

function file_content_xml([\$xml_path = WS_BOOKS], [\$node_path = UNDEFINED], [\$lng = UNDEFINED]) *[line 37]*

Function Parameters:

- **\$xml_path**
- **\$node_path**
- **\$lng**

Returns the path of the content.xml for the given node and language.

function get_arr_navigation(\$node_path, [\$cache_path = UNDEFINED], [\$lng = UNDEFINED]) *[line 118]*

Function Parameters:

- **\$node_path**
- **\$cache_path**
- **\$lng**

Reads from cache 'navigation.txt' for the given node and returns an associative array with the template variables this_path, next_path, prev_path, up_path, this_title, next_title, prev_title, up_title.

function get_date_str(\$timestamp) *[line 320]*

Function Parameters:

- **\$timestamp**

Converts the given time stamp to a date string.

If the time is today, then give just hours and minutes, otherwise give more information.

function get_node_state() *[line 217]*

Reads the file state.txt of the current node and returns

an associative array with the variables found in it. This file contains two lines. The first line is the lock line and contains 'lock:mode:l_user:l_email:l_timestamp', where lock is 'locked' or 'unlocked' and mode is 'edit' or 'approve'. The second line is the modification line and it contains 'status:m_user:m_email:m_timestamp', where status is 'synchronized', or 'modified'. The keys of the associative array are: lock, mode, l_user, l_email, l_timestamp, status, m_user, m_email, m_timestamp.

function get_state_filename() *[line 196]*

Returns the file state.txt of the current node.

function is_locked(\$arr_state) *[line 299]*

Function Parameters:

- **\$arr_state**

Returns true if the node is locked and the lock has not expired.

`function locked_by_somebody([$arr_state = UNDEFINED]) [line 291]`

Function Parameters:

- **\$arr_state**

Returns true and displays a notification message, if the node is locked by another user.

`function process_index($transformer, [$arr_params = array()], [$books = UNDEFINED], [$lng = UNDEFINED]) [line 81]`

Function Parameters:

- **\$transformer**
- **\$arr_params**
- **\$books**
- **\$lng**

applies the given transformer to index.xml

`function set_node_lock($lock, $mode) [line 270]`

Function Parameters:

- **\$lock**
- **\$mode**

Lock or unlock the current node.

The **\$lock** parameter can be 'locked' or 'unlocked', and **\$mode** can be 'edit' or 'approve'.

function set_node_status(\$new_status) [*line 255*]

Function Parameters:

- **\$new_status**

Set the status of the current node to the given new status.

The new status can be: synchronized or modified.

function update_cache([\$public = 'workspace'], [\$node_path = UNDEFINED], [\$lng = UNDEFINED]) [*line 159*]

Function Parameters:

- **\$public**
- **\$node_path**
- **\$lng**

Update the cache file content.html from content.xml

function write_node_state(\$arr_state) [*line 242*]

Function Parameters:

- **\$arr_state**

langs.php

- **Package** docbook

`include_once TPL.'languages/func.languages.php'` [*line 25*]

Package docbook Classes

Class docbook

[line 32]

Handles the content of the book, editing, approving, navigation, etc.

- **Package** docbook

void function docbook::add_accessright_vars() [line 92]

add variables about access rights of the user

void function docbook::get_button_vars() [line 151]

Returns an associative array with keys 'edit', 'approve' and 'admin' and whith values 'true' or 'false', according to the access rights of the current user.

void function docbook::get_edit_rights() [line 201]

Reads the user edit rights file 'admin/access_rights/username'

which contains lines like this: allow:edit,approve:node_path_1,node_path_2:lng1,lng2
deny:approve:node_path_3,node_path_4:lng2 After matching the current node and language with these lines, returns array(\$edit, \$approve) with values 'true' or 'false'.

void function docbook::init() [line 34]

void function docbook::is_admin() [line 104]

Returns 'true' if the current user is an admin for the current book, otherwise returns 'false'.

void function docbook::lang_match(\$lang_list) [line 262]

Function Parameters:

- **\$lang_list**

Returns true if one of the language ids in the list matches the current language, otherwise returns false. The lang_list is a comma separated list of languages, or 'ALL'.

'ALL' matches any language.

void function docbook::node_match(\$node_list) [line 243]

Function Parameters:

- **\$node_list**

Returns true if one of the node_path expressions in the list matches the current node path, otherwise returns false.

The node_list is a comma separated list of node_path expressions (which are actually regular expressions), or is the string 'ALL'. 'ALL' matches any node path.

void function docbook::onParse() [line 82]

void function docbook::onRender() [line 121]

void function docbook::on_set_lng(\$event_args) [line 71]

Function Parameters:

- **\$event_args**

void function docbook::on_set_mode(\$event_args) [line 56]

Function Parameters:

- **\$event_args**

void function docbook::on_set_node(\$event_args) [line 43]

Function Parameters:

- **\$event_args**

void function docbook::on_set_node_id(\$event_args) [line 49]

Function Parameters:

- **\$event_args**

Class langs

[line 32]

A list of languages in which the book is available.

- **Package** docbook

void function langs::onRender() [line 34]

Class view

[line 30]

Display the cache (html) file for the current node of the book.

- **Package** docbook

void function view::init() [line 32]

void function view::onParse() [line 37]

void function view::setup_webnotes() [line 54]

set proper values to the parameters of the 'webnotes' component

func.popup_window.php

Open a popup window from the JS code, which has the given title and displays the given message.

- **Package** docbook
- **Sub-Package** admin

tags.php

- **Package** docbook
- **Sub-Package** admin

include_once **__FILE__** *[line 25]*

Class admin

[line 32]

Some administration tasks about the book.

Can be accessed only by an admin of the book.

- **Package** docbook
- **Sub-Package** admin

void function admin::add_rs_modified_nodes() *[line 97]*

add to webPage the recordset modified_nodes

void function admin::get_title(\$book_id, \$lng, \$node_path) *[line 122]*

Function Parameters:

- **\$book_id**
- **\$lng**
- **\$node_path**

void function admin::onRender() *[line 91]*

void function admin::on_update_booklist(\$event_args) [line 86]

Function Parameters:

- **\$event_args**

void function admin::on_update_downloadables(\$event_args) [line 59]

Function Parameters:

- **\$event_args**

void function admin::on_update_search_index(\$event_args) [line 34]

Function Parameters:

- **\$event_args**

Class tags

[line 34]

Handle the tags of a book (create new tag, list existing tags, fix book to a tag, etc.)

- **Package** docbook
- **Sub-Package** admin

void function tags::add_rs_tags() [line 192]

Add to webpage a recordset of tags.

void function tags::get_book_dir() [line 229]

return the directory of the book: content/books/xml/book_id/lng/

void function tags::get_current_tag() [line 140]

void function tags::get_head_revision() [line 172]

void function tags::get_revision_date() [*line 162*]

void function tags::get_revision_nr() [*line 153*]

void function tags::get_svn_url() [*line 214*]

return the URL of the SVN repository: file://path/book_id/Ing

void function tags::is_fixed() [*line 182*]

void function tags::onRender() [*line 130*]

void function tags::on_add_tag(\$event_args) [*line 108*]

Function Parameters:

- **\$event_args**

void function tags::on_del_tag(\$event_args) [*line 122*]

Function Parameters:

- **\$event_args**

void function tags::on_set_date(\$event_args) [*line 101*]

Function Parameters:

- **\$event_args**

void function tags::on_set_rev(\$event_args) [*line 87*]

Function Parameters:

- **\$event_args**

void function tags::on_set_tag(\$event_args) [*line 68*]

Function Parameters:

- **\$event_args**

void function tags::set_revision(\$revision) [*line 36*]

Function Parameters:

- **\$revision**

void function tags::write_fixed_file(\$content) [*line 62*]

Function Parameters:

- **\$content**

Write the file 'fixed' which prevents the update of the latest changes of the book in the public copy.

approve.php

- **Package** docbook
- **Sub-Package** approve

include_once **DOCBOOK**.'/edit/edit_funcs.php' *[line 29]*

include_once **DOCBOOK**.'/edit/content/process_content.php' *[line 26]*

Class approve

[line 38]

Review and approve (or cancel) the modifications that are done to the current node of the book.

- **Package** docbook
- **Sub-Package** approve

void function approve::get_content_diff() *[line 268]*

get the modifications of content.xml

void function approve::get_svn_status() *[line 320]*

Get the modifications made in media files
(added/replaced/deleted images etc.)

void function approve::msg_book_fixed_to_tag(\$tag) *[line 161]*

Function Parameters:

- **\$tag**

void function approve::non_recursive_commit(\$log) [line 117]

Function Parameters:

- **\$log**

void function approve::non_recursive_revert() [line 55]

void function approve::onRender() [line 215]

void function approve::on_commit(\$event_args) [line 106]

Function Parameters:

- **\$event_args**

void function approve::on_revert(\$event_args) [line 46]

Function Parameters:

- **\$event_args**

void function approve::on_set_lock(\$event_args) [line 40]

Function Parameters:

- **\$event_args**

void function approve::recursive_commit(\$log) [line 176]

Function Parameters:

- **\$log**

void function approve::recursive_revert() [line 77]

void function approve::set_all_synchronized(\$node_dir) [line 101]

Function Parameters:

- **\$node_dir**

Class revisions

[line 31]

Handle (view, select) the previous revisions of the book.

- **Package** docbook
- **Sub-Package** approve

void function revisions::add_file_revision() [line 155]

add the variable `{{file_revision}}`

void function revisions::add_rev_diff() [line 212]

add `{{rev_diff}}`, the difference
of the selected revision with the HEAD

void function revisions::add_rs_revs() [line 131]

Add to webpage a recordset of revs.

void function revisions::add_rs_tags() [line 109]

Add to webpage a recordset of tags.

void function revisions::clear() [line 51]

void function revisions::get_revision() [line 182]

return the content.xml at the selected revision (or date, or tag)

void function revisions::get_svn_url() [line 297]

return the URL of the SVN repository: `file://path/book_id/Inq`

void function revisions::init() [line 33]

void function revisions::onRender() [line 96]

void function revisions::on_hide(\$event_args) [line 46]

Function Parameters:

- **\$event_args**

void function revisions::on_save(\$event_args) [*line 79*]

Function Parameters:

- **\$event_args**

void function revisions::on_set_date(\$event_args) [*line 72*]

Function Parameters:

- **\$event_args**

void function revisions::on_set_rev(\$event_args) [*line 65*]

Function Parameters:

- **\$event_args**

void function revisions::on_set_tag(\$event_args) [*line 58*]

Function Parameters:

- **\$event_args**

void function revisions::on_show(\$event_args) [*line 41*]

Function Parameters:

- **\$event_args**

convert_from_xml.php

- **Package** docbook
- **Sub-Package** edit

function xml_to_html(\$xml_file) [*line 30*]

Function Parameters:

- **\$xml_file**

function xml_to_text(\$xml_file) [*line 41*]

Function Parameters:

- **\$xml_file**

convert_to_xml.php

- **Package** docbook
- **Sub-Package** edit

function get_id_title() [*line 32*]

returns the id and the title of the current node

function html_to_xml(\$content) [*line 94*]

Function Parameters:

- **\$content**

function is_wellformed(\$xml_content) [*line 71*]

Function Parameters:

- **\$xml_content**

Check that the given XML string is well formed.

Return true if well formed, false otherwise.

function latex_to_xml(\$content) [*line 176*]

Function Parameters:

- **\$content**

function texi_to_xml(\$content) [*line 186*]

Function Parameters:

- **\$content**

function text_to_xml(\$content) [*line 131*]

Function Parameters:

- **\$content**

`function xml_to_xml($content) [line 50]`

Function Parameters:

- **\$content**

edit_content.php

- **Package** docbook
- **Sub-Package** edit

include_once __FILE__ *[line 29]*

include_once __FILE__ *[line 28]*

include_once __FILE__ *[line 27]*

include_once __FILE__ *[line 26]*

include_once __FILE__ *[line 25]*

func.popup_window.php

Open a popup window from the JS code, which has the given title and displays the given message.

- **Package** docbook
- **Sub-Package** edit

function popup_window(\$title, \$message) [*line 32*]

Function Parameters:

- **\$title**
- **\$message**

process_content.php

This file contains functions about processing the content of a node that are used by the edit_content webbox and by other modules.

- **Package** docbook
- **Sub-Package** edit

function compact_cdata(\$content) *[line 190]*

Function Parameters:

- **\$content**

Replace explicit CDATA like this:

--tag <![CDATA[abc xyz]]> ---- by a compact (implicit) CDATA like this: --tag--
abc xyz ----

Also, replace <![CDATA[xyz]]> by [[xyz]]

function expand_cdata(\$content) *[line 151]*

Function Parameters:

- **\$content**

Replace implicit CDATA like this:

--tag-- abc xyz ---- by an explicit CDATA like this: --tag <![CDATA[abc xyz]]> ----

Also, replace [[xyz]] by <![CDATA[xyz]]>

function get_node_content(\$xml_file, \$mode) *[line 39]*

Function Parameters:

- **\$xml_file**

- **\$mode**

Builds the content of the node by processing content.xml with xml2mode.xsl, and assigns it to a template variable.

include_once **__FILE__** *[line 33]*

function putback_cdata(\$str, [\$mode = 'xml']) *[line 121]*

Function Parameters:

- **\$str**
- **\$mode**

replace <CDATA>x</CDATA> by the corresponding <![CDATA[...]]>

function putback_comments(\$str) *[line 110]*

Function Parameters:

- **\$str**

replace <comment>x</comment> by the corresponding <!--...-->

function strip_cdata(\$str) *[line 76]*

Function Parameters:

- **\$str**

replace <![CDATA[...]]> by <CDATA>x</CDATA>

function strip_comments(\$str) *[line 93]*

Function Parameters:

- **\$str**

replace `<!--...-->` by `<comment>x</comment>`

validate.php

- **Package** docbook
- **Sub-Package** edit

function display_error_messages(&\$xml_content, &\$err_lines) [*line 78*]

Function Parameters:

- **&\$xml_content**
- **&\$err_lines**

display a popup window with error messages

function get_short_xml_line(\$xml_line, \$elem) [*line 112*]

Function Parameters:

- **\$xml_line**
- **\$elem**

extract from the xml line a part around the error

function validate_xml(\$xml_content) [*line 30*]

Function Parameters:

- **\$xml_content**

edit.php

- **Package** docbook
- **Sub-Package** edit

include_once __FILE__ *[[line 25]]*

edit_funcs.php

This function sets a lock before index (structure of the book)

is modified, in order to make sure that the change in the index and update of the cache files (navigation and subnodes) happens as a single transaction, and concurrent editing does not have unexpected results. Since the structure of the book is the same for all the languages and the changes are synchronized for all the languages, the lock locks all the languages.

- **Package** docbook
- **Sub-Package** edit

function add_subsections_rs() [*line 118*]
add the recordset 'subsections'

function add_to_modified_nodes([\$path = UNDEFINED]) [*line 62*]
Function Parameters:

- **\$path**

Append the node path to the file 'modified_nodes.txt'

function changeid_movefolders(\$node_path, \$new_node_path) [*line 496*]
Function Parameters:

- **\$node_path**
- **\$new_node_path**

move folders to the new node path, for all the languages

function changeid_setid(\$new_id) [*line 539*]
Function Parameters:

- **\$new_id**

set the new id in content.xml, for all the languages

function changeid_updatecache(\$node_path) *[line 572]*

Function Parameters:

- **\$node_path**

function change_id_validate(\$new_id) *[line 445]*

Function Parameters:

- **\$new_id**

Returns true if the new id is OK, otherwise returns false.

function create_new_node(\$id, \$title, \$type) *[line 387]*

Function Parameters:

- **\$id**
- **\$title**
- **\$type**

create a new node and node files

function delete_removefolders(\$node_path, \$lng) *[line 359]*

Function Parameters:

- **\$node_path**
- **\$lng**

function delete_removefolders_all_langs(\$node_path) *[line 348]*

Function Parameters:

- **\$node_path**

function get_node_references() *[line 607]*

Find the nodes that reference this one and return a list of them as a string (having a line for each node). The return value is empty string (") if there are no referencies.

This function is used when a node is deleted or when the node id is changed, in order to make sure that there are no dangling referencies in the document. Otherwise the document will become invalid and will fail to convert to other formats.

function lock_index() *[line 38]*

function process_index_node(\$transformer, [\$node_path = UNDEFINED]) *[line 139]*

Function Parameters:

- **\$transformer**
- **\$node_path**

Applies the given transformer to index.xml, passing the node as a parameter, and returns the output.

function remove_from_modified_nodes([\$path = UNDEFINED], [\$recursive = false]) *[line 89]*

Function Parameters:

- **\$path**
- **\$recursive**

Remove the node path from the file 'modified_nodes.txt'

function transform_index(\$action, [\$arr_params = array()]) *[line 157]*

Function Parameters:

- **\$action**
- **\$arr_params**

Applies a transformation to index.xml and then writes the result back to index.xml.

function unlock_index() [*line 54*]
function updatenavig_movedown(\$node_path) [*line 313*]

Function Parameters:

- **\$node_path**

function updatenavig_moveup(\$node_path) [*line 302*]

Function Parameters:

- **\$node_path**

function update_cache_files(\$node_path, [\$lng = UNDEFINED]) [*line 335*]

Function Parameters:

- **\$node_path**
- **\$lng**

function update_cache_files_all_langs(\$node_path) [*line 324*]

Function Parameters:

- **\$node_path**

function update_navigation(\$node_path, \$lng) [*line 215*]

Function Parameters:

- **\$node_path**
- **\$lng**

Update the file navigation.txt for the given node and language.

function update_navigation_all_langs(\$node_path) [*line 203*]

Function Parameters:

- **\$node_path**

Update the files navigation.txt for all languages, for the given node.

```
function update_subnodes_html($node_path, $lng, [$recursive = 'true']) [line 264]
```

Function Parameters:

- **\$node_path**
- **\$lng**
- **\$recursive**

Updates the cache files 'subnodes.html' for the given language, for the given node and all its ancestors, by applying 'update-subnodes.xsl' to index.xml.

If a third parameter is given and its value is not 'true', then only the subnodes of the given node are generated (without ascending to the ancestors).

```
function update_subnodes_html_all_langs([$node_path = UNDEFINED]) [line 239]
```

Function Parameters:

- **\$node_path**

Updates the cache files subnodes.html for the given node, for all its ancestors, and for all the languages.

Class edit

[line 33]

Edit the content of the current node.

- **Package docbook**

- **Sub-Package** edit

void function edit::add_nodetypes_rs() [line 270]

Create and add to webPage the recordset 'node_types' with the types of the nodes that can be added to the current node (preface, appendix, chapter, section, simplesect).

void function edit::onParse() [line 208]

void function edit::onRender() [line 217]

void function edit::on_add_subnode(\$event_args) [line 110]

Function Parameters:

- **\$event_args**

void function edit::on_change_id(\$event_args) [line 144]

Function Parameters:

- **\$event_args**

void function edit::on_change_title(\$event_args) [line 179]

Function Parameters:

- **\$event_args**

void function edit::on_delete(\$event_args) [line 61]

Function Parameters:

- **\$event_args**

void function edit::on_move_down(\$event_args) [line 51]

Function Parameters:

- **\$event_args**

void function edit::on_move_up(\$event_args) [line 41]

Function Parameters:

- **\$event_args**

void function edit::on_set_lock(\$event_args) [line 35]

Function Parameters:

- **\$event_args**

Class edit_content

[line 35]

- **Package** docbook
- **Sub-Package** edit

void function edit_content::add_tab_items() [line 142]

Add to the webpage a recordset with the items of the tabs (modes).

void function edit_content::error_on_save(\$content) [line 113]

Function Parameters:

- **\$content**

void function edit_content::init() [line 37]

void function edit_content::onRender() [line 128]

void function edit_content::on_save(\$event_args) [line 49]

Function Parameters:

- **\$event_args**

void function edit_content::on_set_mode(\$event_args) [line 43]

Function Parameters:

- **\$event_args**

Class edit_media

[line 31]

Add, rename, delete the media files of the node.

- **Package** docbook
- **Sub-Package** edit

void function edit_media::get_media_path() [line 122]

void function edit_media::onRender() [line 88]

void function edit_media::on_delete(\$event_args) [line 54]

Function Parameters:

- **\$event_args**

void function edit_media::on_rename(\$event_args) [line 65]

Function Parameters:

- **\$event_args**

void function edit_media::on_upload(\$event_args) [line 33]

Function Parameters:

- **\$event_args**

package.WikiConverter.php

- **Package** docbook
- **Sub-Package** wikiconverter

include_once [\\$path.'class.ListItem.php'](#)*[line 37]*

include_once [\\$path.'class.WikiParser.php'](#)*[line 38]*

include_once [\\$path.'fun.regex_replace.php'](#)*[line 39]*

include_once [\\$path.'class.Listing.php'](#)*[line 36]*

include_once [\\$path.'class.Block.php'](#)*[line 35]*

include_once [\\$path.'class.Tpl.php'](#)*[line 32]*

include_once [\\$path.'class.Text.php'](#)*[line 33]*

include_once [\\$path.'class.Paragraph.php'](#)*[line 34]*

include_once [\\$path.'class.Info.php'](#)*[line 31]*

Class Block

[line 29]

- **Package** docbook
- **Sub-Package** wikiconverter

Block::\$block_title

mixed = [line 32]

Block::\$block_type

mixed = [line 31]

Constructor *void* function `Block::Block([$type = 'code'], [$title = ''])` *[line 34]*

Function Parameters:

- **\$type**
- **\$title**

void function `Block::line_to_html($line, $indent)` *[line 53]*

Function Parameters:

- **\$line**
- **\$indent**

void function `Block::line_to_xml($line, $indent)` *[line 129]*

Function Parameters:

- **\$line**
- **\$indent**

void function `Block::to_html($indent, $class)` *[line 44]*

Function Parameters:

- **\$indent**
- **\$class**

void function `Block::to_xml($indent)` *[line 60]*

Function Parameters:

- **\$indent**

Class Info

[line 31]

Handle bookinfo or articleinfo elements.

- **Package** docbook
- **Sub-Package** wikiconverter

Info::\$abstract

mixed = [line 34]

Info::\$authors

mixed = [line 33]

Info::\$copyright

mixed = [line 36]

Info::\$keywords

mixed = [line 35]

Constructor *void* function Info::Info() [line 38]

void function Info::parse(\$content) [line 50]

Function Parameters:

- **\$content**

Extract the various parts of the info section

(@author, @abstract, etc.)

void function Info::parse_abstract(\$str) [line 103]

Function Parameters:

- **\$str**

void function Info::parse_author(\$str) [line 81]

Function Parameters:

- **\$str**

void function Info::parse_copyright(\$str) [line 108]

Function Parameters:

- **\$str**

void function Info::parse_dummy(\$str) [line 79]

Function Parameters:

- **\$str**

void function Info::parse_keywords(\$str) [line 92]

Function Parameters:

- **\$str**

void function Info::render_abstract() [line 154]

void function Info::render_authors() [line 123]

void function Info::render_copyright() [line 178]

void function Info::render_keywords() [line 163]

void function Info::to_xml() [line 114]

convert the info data to xml

Class Listing

[line 29]

- **Package** docbook
- **Sub-Package** wikiconverter

Listing::\$bullet

mixed = [line 32]

Listing::\$indent

mixed = [line 31]

Constructor *void* function Listing::Listing([\$indent = "], [\$bullet = '*']) *[line 34]*

Function Parameters:

- **\$indent**
- **\$bullet**

void function Listing::to_html(\$indent, \$class) *[line 44]*

Function Parameters:

- **\$indent**
- **\$class**

void function Listing::to_xml(\$indent) *[line 66]*

Function Parameters:

- **\$indent**

Class ListItem *[line 29]*

- **Package** docbook

- **Sub-Package** wikiconverter

Constructor *void* function ListItem::ListItem() [*line 31*]
void function ListItem::to_html(\$indent, \$class) [*line 39*]

Function Parameters:

- **\$indent**
- **\$class**

void function ListItem::to_xml(\$indent) [*line 48*]

Function Parameters:

- **\$indent**

Class Paragraph

[*line 29*]

- **Package** docbook
- **Sub-Package** wikiconverter

Constructor *void* function Paragraph::Paragraph() [*line 31*]
void function Paragraph::to_html(\$indent, \$class) [*line 39*]

Function Parameters:

- **\$indent**
- **\$class**

void function Paragraph::to_xml(\$indent) [*line 47*]

Function Parameters:

- **\$indent**

Class Text

[line 29]

- **Package** docbook
- **Sub-Package** wikiconverter

Constructor *void* function Text::Text() *[line 31]*
void function Text::line_to_html(\$line, \$indent) *[line 37]*

Function Parameters:

- **\$line**
- **\$indent**

void function Text::line_to_xml(\$line, \$indent) *[line 44]*

Function Parameters:

- **\$line**
- **\$indent**

Class Tpl

[line 29]

- **Package** docbook

- **Sub-Package** wikiconverter

Tpl::\$contents

mixed = [line 33]

Tpl::\$id

mixed = [line 31]

Tpl::\$type

mixed = [line 32]

Constructor *void* function Tpl::Tpl([\$type = 'Template']) *[line 36]*

Function Parameters:

- **\$type**

Constructor

void function Tpl::get_subtemplates() *[line 60]*

Returns an array of ids of the templates that are referenced from the contents of this template.

void function Tpl::get_tpl_id(\$line) *[line 50]*

Function Parameters:

- **\$line**

If the given line is a reference to another template, returns the id of the referenced template, otherwise return empty string "".

void function Tpl::line_to_html(\$line, \$indent) *[line 96]*

Function Parameters:

- **\$line**
- **\$indent**

void function Tpl::line_to_xml(\$line, \$indent) [line 127]

Function Parameters:

- **\$line**
- **\$indent**

void function Tpl::to_html([\$indent = "], [\$class = "]) [line 72]

Function Parameters:

- **\$indent**
- **\$class**

void function Tpl::to_html_table() [line 134]

Debug function.

void function Tpl::to_xml([\$indent = "]) [line 102]

Function Parameters:

- **\$indent**

Class WikiParser

[line 29]

- **Package** docbook
- **Sub-Package** wikiconverter

WikiParser::\$arr_lines

mixed = [line 32]

lines that will be parsed and their index

WikiParser::\$idx

mixed = [line 33]

Constructor *void* function WikiParser::WikiParser() [*line 35*]

void function WikiParser::append_line(&\$tpl, [\$line = 'UNDEFINED']) [*line 296*]

Function Parameters:

- **&\$tpl**
- **\$line**

append the current line to the given template

void function WikiParser::block_end() [*line 334*]

void function WikiParser::block_start() [*line 324*]

void function WikiParser::empty_line() [*line 312*]

void function WikiParser::end_of_lines() [*line 283*]

return true if the index is at the end of the lines

void function WikiParser::get_bullet(\$mark) [*line 356*]

Function Parameters:

- **\$mark**

return the bullet of the given mark

void function WikiParser::get_current_line() [*line 276*]

get the current line that is being parsed

void function WikiParser::get_indentation(\$mark) [*line 349*]

Function Parameters:

- **\$mark**

return the indentaion of the given mark

void function WikiParser::get_listitem_mark() [*line 340*]
void function WikiParser::link_template(&\$parent, \$child) [*line 303*]

Function Parameters:

- **&\$parent**
- **\$child**

add a link from the parent template to the child template

void function WikiParser::listitem_end(\$mark) [*line 450*]

Function Parameters:

- **\$mark**

void function WikiParser::listitem_start(\$mark) [*line 435*]

Function Parameters:

- **\$mark**

void function WikiParser::list_end(\$mark) [*line 392*]

Function Parameters:

- **\$mark**

void function WikiParser::list_start([\$mark = ""]) [*line 363*]

Function Parameters:

- **\$mark**

return true if the current line starts a new list

void function WikiParser::next_line() [*line 290*]

move the index to the nex line

void function WikiParser::nonempty_line() [*line 318*]

void function WikiParser::parse_block() [line 139]
void function WikiParser::parse_file(\$filename) [line 44]

Function Parameters:

- **\$filename**

void function WikiParser::parse_info() [line 176]
void function WikiParser::parse_lines(\$arr_lines) [line 69]

Function Parameters:

- **\$arr_lines**

void function WikiParser::parse_list() [line 216]
void function WikiParser::parse_listitem(\$mark) [line 237]

Function Parameters:

- **\$mark**

void function WikiParser::parse_paragraph() [line 110]
void function WikiParser::parse_string(\$str) [line 57]

Function Parameters:

- **\$str**

void function WikiParser::parse_text() [line 82]

parse the text wiki

void function WikiParser::template_list() [line 498]

(debug) Outputs the data of each template in \$tpl_collection.

void function WikiParser::to_tree(\$tpl, \$indent) [line 537]

Function Parameters:

- **\$tpl**
- **\$indent**

Returns the structure of loaded templates as a tree.

void function WikiParser::tpl_to_tree() [*line 523*]

(debug) Returns the structure of the loaded templates as a tree.

fun.regex_replace.php

Replace wiki inline markups with docbook markups, using regular expressions.

- **Package** docbook
- **Sub-Package** wikiconverter--

function convert_menu_items(\$str) [*line 182*]

Function Parameters:

- **\$str**

This function gets a string like this: 'M_enu->S_ubmenu->MenuI_tem (Ctrl-C)' and returns a docbook code like this:

```
<menuchoice>                                <shortcut>
<keycombo><keycap>Ctrl</keycap><keycap>C</keycap></keycom
bo>  </shortcut>  <guimenu><accel>M</accel>enu</guimenu>
<guisubmenu><accel>S</accel>ubmenu</guisubmenu>
<guimenuitem>Menu<accel>I</accel>tem</guimenuitem>
</menuchoice>.
```

function mediaobject(\$filename, [\$width = "], [\$salt = "]) [*line 253*]

Function Parameters:

- **\$filename**
- **\$width**
- **\$salt**

Returns a <mediobject> element

function regex_replace(\$line) [*line 54*]

Function Parameters:

- **\$line**

Package webobjects Procedural Elements

tabs1_items.php

The `$menu_items` array contains the items of the tabs1.

- **Package** webobjects
- **Sub-Package** tabs

Package webobjects Classes

Class tabs1 *[line 29]*

- **Package** webobjects
- **Sub-Package** tabs

void function tabs1::add_items_list() *[line 54]*

void function tabs1::init() *[line 31]*

void function tabs1::onRender() *[line 49]*

void function tabs1::on_select(\$event_args) *[line 43]*

Function Parameters:

- **\$event_args**

Appendices

Appendix A - Class Trees

Package DocBookWiki

books

- WebObject
 - [books](#)

help

- WebObject
 - [help](#)

languages

- WebObject
 - [languages](#)

main

- WebObject
 - [main](#)

search

- WebObject
 - [search](#)

Package docbook

admin

- WebObject
 - [admin](#)

approve

- WebObject
 - [approve](#)

docbook

- WebObject
 - [docbook](#)

edit

- WebObject
 - [edit](#)

edit_content

- WebObject
 - [edit_content](#)

edit_media

- WebObject
 - [edit_media](#)

Info

- [Info](#)

langs

- WebObject
 - [langs](#)

revisions

- WebObject
 - [revisions](#)

tags

- WebObject
 - [tags](#)

Tpl

- [Tpl](#)
 - [Block](#)
 - [Listing](#)
 - [ListItem](#)
 - [Paragraph](#)

- [Text](#)

view

- WebObject
 - [view](#)

WikiParser

- [WikiParser](#)

Package admin

addbook

- WebObject
 - [addbook](#)

admin

- WebObject
 - [admin](#)

books

- WebObject
 - [books](#)

book_list

- WebObject
 - [book_list](#)

edit_menu

- WebObject
 - [edit_menu](#)

edit_rights

- WebObject
 - [edit_rights](#)

import_doc

- WebObject
 - [import_doc](#)

modified_nodes

- WebObject
 - [modified_nodes](#)

uEdit

- WebObject
 - [uEdit](#)

uList

- WebObject
 - [uList](#)

userEdit

- formWebObj
 - [userEdit](#)

userFilter

- WebObject
 - [userFilter](#)

userList

- WebObject
 - [userList](#)

user_data

- WebObject
 - [user_data](#)

Package webobjects

tabs1

- WebObject
 - [tabs1](#)

Index

A

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| approve::msg_book_fixed_to_tag() | 52 |
| approve::non_recursive_commit() | 53 |
| approve::non_recursive_revert() | 53 |
| approve::get_svn_status() | 52 |
| <i>Get the modifications made in media files</i> | |
| approve::get_content_diff() | 52 |
| <i>get the modifications of content.xml</i> | |
| approve.php | 52 |
| approve | 52 |
| <i>Review and approve (or cancel) the modifications that are done to the current node of the book.</i> | |
| approve::onRender() | 53 |
| approve::on_commit() | 53 |
| approve::set_all_synchronized() | 53 |
| add_subsections_rs() | 66 |
| <i>add the recordset 'subsections'</i> | |
| add_to_modified_nodes() | 66 |
| <i>Append the node path to the file 'modified_nodes.txt'</i> | |
| approve::recursive_revert() | 53 |
| approve::recursive_commit() | 53 |
| approve::on_revert() | 53 |
| approve::on_set_lock() | 53 |
| admin::on_update_search_index() | 49 |
| admin::on_update_downloadables() | 49 |
| addbook::get_initial_xml() | 5 |
| <i>constructs and returns an initial xml document</i> | |
| addbook::get_legalnotice() | 5 |
| addbook::keywords_to_xml() | 5 |
| <i>Gets a comma separated list of keywords: 'k1, k2, k3,'</i> | |
| addbook::author_to_xml() | 5 |
| <i>Gets a string like this: 'Surname, Firstname, email'</i> | |
| addbook | 4 |
| <i>This module is used to import a new book/article in the system or to create a new one. A list of the current books in the system is displayed as well and it is also possible to remove some books from the system.</i> | |
| admin::init() | 4 |
| admin::onParse() | 4 |
| addbook::on_create() | 5 |
| APP_CONFIG_FILE | 26 |
| <i>The file that contains additional configuration options, which are specific for the application.</i> | |
| admin::onRender() | 48 |
| admin::on_update_booklist() | 49 |

| | |
|-----------------------------------------------------------------------------------------------------------------------|----|
| admin::get_title() | 48 |
| admin::add_rs_modified_nodes() | 48 |
| <i>add to webPage the recordset modified_nodes</i> | |
| ADMIN | 28 |
| admin | 48 |
| <i>Some administration tasks about the book.</i> | |
| admin | 4 |
| <i>Modules and interfaces that are used by the superuser and by the book admins. Is accessed through 'admin.php'.</i> | |

B

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------|----|
| book fixed to tag() | 39 |
| <i>Check whether the public copy of the book is fixed to a certain tag or revision and return it. If book is not fixed, return false.</i> | |
| Block | 74 |
| books::onRender() | 32 |
| books | 32 |
| <i>A listbox of the books/articles in the system.</i> | |
| BOOKS | 28 |
| Block::\$block_title | 75 |
| Block::\$block_type | 75 |
| Block::to_xml() | 75 |
| Block::to_html() | 75 |
| Block::line_to_xml() | 75 |
| Block::line_to_html() | 75 |
| books::on_select() | 8 |
| books::onRender() | 8 |
| book_list::on_delete() | 6 |
| book_list::onRender() | 6 |
| book_list::init() | 6 |
| book_list::add_recordsets() | 6 |
| book_list::on_set_visible() | 6 |
| books | 8 |
| <i>The books that the user can administrate.</i> | |
| books::init() | 8 |
| books::get_book_list_arr() | 8 |
| books::add_book_title() | 8 |
| books::add_book_list_rs() | 8 |
| book_list | 6 |
| <i>Displays a list of the current books in the system.</i> | |

C

| | |
|------------------------------------------------|----|
| constructor Info::Info() | 76 |
| constructor Listing::Listing() | 78 |
| constructor Block::Block() | 75 |
| create_new_node() | 67 |
| <i>create a new node and node files</i> | |
| change_id_validate() | 67 |

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <i>Returns true if the new id is OK, otherwise returns false.</i> | |
| constructor ListItem::ListItem() | 79 |
| constructor Paragraph::Paragraph() | 79 |
| convert_menu_items() | 87 |
| <i>This function gets a string like this: 'M_enu->S_ubmenu->MenuI_tem (Ctrl-C)' and returns a docbook code like this:</i> | |
| constructor WikiParser::WikiParser() | 83 |
| constructor Tpl::Tpl() | 81 |
| <i>Constructor</i> | |
| constructor Text::Text() | 80 |
| changeid_updatecache() | 67 |
| changeid_setid() | 66 |
| <i>set the new id in content.xml, for all the languages</i> | |
| CACHE | 28 |
| const.Paths.php | 28 |
| \$const | 26 |
| const.Options.php | 26 |
| <i>The constants defined in this file change the behaviour of the framework. You can change the values of the constants according to the instructions given in comments.</i> | |
| CONTENT | 28 |
| CSS | 28 |
| changeid_movefolders() | 66 |
| <i>move folders to the new node path, for all the languages</i> | |
| compact_cdata() | 61 |
| <i>Replace explicit CDATA like this:</i> | |
| convert_to_xml.php | 57 |
| convert_from_xml.php | 56 |
| const.Debug.php | 24 |
| <i>The following constants are debug constants.</i> | |

D

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| docbook::onParse() | 45 |
| docbook::onRender() | 45 |
| docbook::node_match() | 45 |
| <i>Returns true if one of the node_path expressions in the list matches the current node path, otherwise returns false.</i> | |
| docbook::lang_match() | 45 |
| <i>Returns true if one of the language ids in the list matches the current language, otherwise returns false. The lang_list is a comma separated list of languages, or 'ALL'.</i> | |
| docbook::is_admin() | 44 |
| <i>Returns 'true' if the current user is an admin for the current book, otherwise returns 'false'.</i> | |
| docbook::on_set_lng() | 45 |
| docbook::on_set_mode() | 45 |
| delete_removefolders() | 67 |
| delete_removefolders_all_langs() | 67 |
| display_error_messages() | 64 |
| <i>display a popup window with error messages</i> | |
| docbook::on_set_node_id() | 46 |

| | |
|------------------------------------------------------------------------------|----|
| docbook::on_set_node() | 45 |
| docbook::init() | 44 |
| docbook::get_edit_rights() | 44 |
| <i>Reads the user edit rights file 'admin/access_rights/username'</i> | |
| DEBUG_TEMPLATES | 24 |
| <i>if this constant is true, the framework outputs the tree</i> | |
| DEBUG_WEBOBJECTS | 25 |
| <i>if this constant is true, the framework outputs a box</i> | |
| DEBUG_STATEVARS | 24 |
| <i>if this constant is true, the framework outputs the state</i> | |
| DEBUG_RECORDSETS | 24 |
| <i>if this constant is true, the framework outputs all the</i> | |
| DEBUG_INCLUDES | 24 |
| <i>if this constant is true, the framework outputs a box</i> | |
| DB_TYPE | 26 |
| <i>this constant defines the type of DB that the application uses</i> | |
| <i>it can be: "MySQL", "Oracle", "ODBC",</i> | |
| <i>"ProgreSQL", etc.</i> | |
| DISPLAY_CGI_VARS | 26 |
| <i>When this constant is true, then the CGI vars are displayed</i> | |
| <i>at the URL window of the browser. See also SHOW_EXTERNAL_LINK</i> | |
| <i>at const.Debug.php.</i> | |
| docbook::add_accessright_vars() | 44 |
| <i>add variables about access rights of the user</i> | |
| docbook::get_button_vars() | 44 |
| <i>Returns an associative array with keys 'edit', 'approve' and 'admin'</i> | |
| <i>and whith values 'true' or 'false', according to the access rights</i> | |
| <i>of the current user.</i> | |
| docbook | 44 |
| <i>Handles the content of the book, editing, approving, navigation, etc.</i> | |
| docbook.php | 38 |
| DOCBOOK | 28 |
| DEBUG_GOTO | 24 |
| <i>if this constant is true, the framework displays an alert</i> | |

E

| | |
|-------------------------------------------------------------------------------|----|
| edit::on_move_down() | 71 |
| edit::on_delete() | 71 |
| edit::on_move_up() | 71 |
| edit::on_set_lock() | 72 |
| edit_content | 72 |
| edit::on_change_title() | 71 |
| edit::on_change_id() | 71 |
| edit::add_nodetypes_rs() | 71 |
| <i>Create and add to webPage the recordset 'node_types' with the types</i> | |
| <i>of the nodes that can be added to the current node (preface, appendix,</i> | |
| <i>chapter, section, simplesect).</i> | |
| edit::onParse() | 71 |
| edit::onRender() | 71 |
| edit::on_add_subnode() | 71 |
| edit_content::add_tab_items() | 72 |

| | |
|-----------------------------------------------------------------------------------------------------|----|
| <i>Add to the webpage a recordset with the items of the tabs (modes).</i> | |
| edit_content::error_on_save() | 72 |
| edit_media::onRender() | 73 |
| edit_media::on_delete() | 73 |
| edit_media::on_rename() | 73 |
| edit_media::on_upload() | 73 |
| edit_media::get_media_path() | 73 |
| edit_media | 73 |
| <i>Add, rename, delete the media files of the node.</i> | |
| edit_content::init() | 72 |
| edit_content::onRender() | 72 |
| edit_content::on_save() | 72 |
| edit_content::on_set_mode() | 72 |
| edit | 70 |
| <i>Edit the content of the current node.</i> | |
| edit_funcs.php | 66 |
| <i>This function sets a lock before index (structure of the book)</i> | |
| edit_menu::on_delete() | 11 |
| <i>delete the current item and set the parent item as the current one</i> | |
| edit_menu::on_cancel() | 11 |
| <i>discard any modifications and get a copy of the main menu</i> | |
| edit_menu::on_move_down() | 12 |
| <i>move down the given item</i> | |
| edit_menu::on_move_up() | 12 |
| <i>move up the given item</i> | |
| edit_menu::on_paste() | 12 |
| <i>add any items in clipboard as subitems of the current item</i> | |
| edit_menu::on_apply() | 11 |
| <i>apply the modifications to the main menu</i> | |
| edit_menu::on_add_subitem() | 11 |
| <i>add a new subitem to the current item</i> | |
| edit_menu::add_books_rs() | 11 |
| <i>Add a recordset of all the books, which will be displayed in a listbox.</i> | |
| edit_menu::add_subitems_rs() | 11 |
| <i>add a recordset of the subitems of the selected item</i> | |
| edit_menu::init() | 11 |
| edit_menu::onRender() | 11 |
| edit_menu::on_select() | 12 |
| <i>select an item for editing</i> | |
| edit_menu::on_update() | 12 |
| <i>save modifications in id and caption of the current item</i> | |
| EXECUTION_TIME_INFO | 25 |
| <i>if this constant is true, the framework outputs information</i> | |
| edit_content.php | 59 |
| expand_cdata() | 61 |
| <i>Replace implicit CDATA like this:</i> | |
| edit.php | 65 |
| edit_rights::onRender() | 19 |
| edit_rights::init() | 19 |
| edit_menu::select_item() | 13 |
| <i>Get the 'bookid' and the 'caption' of the selected item and refresh the state variables.</i> | |

| | |
|----------------------------------------------------------------------------------------------------------------------------------|----|
| edit_menu::transform() | 13 |
| <i>Applies the given xsl transformer to menu.xml and returns the result. \$arr_params is an associative array of parameters.</i> | |
| edit_menu::transform_menu() | 13 |
| edit_rights | 19 |
| <i>Displays the edit rights of the user.</i> | |
| edit_menu | 10 |
| <i>Modify the menu of the categories and books.</i> | |

F

| | |
|-------------------------------------------------------------------------------------------------------------------------|----|
| func.popup_window.php | 47 |
| <i>Open a popup window from the JS code, which has the given title and displays the given message.</i> | |
| func.popup_window.php | 60 |
| <i>Open a popup window from the JS code, which has the given title and displays the given message.</i> | |
| fun.regex_replace.php | 87 |
| <i>Replace wiki inline markups with docbook markups, using regular expressions.</i> | |
| file_content_xml() | 39 |
| <i>Returns the path of the content.xml for the given node and language.</i> | |
| file_content_html() | 39 |
| <i>Returns the path of the content_lng.html for the given node and language.</i> | |
| func.languages.php | 29 |
| funcs.php | 39 |
| <i>This file has some common functions.</i> | |
| FIRSTPAGE | 26 |
| <i>This is the first page of the application. The framework looks for it at the template folder (specified by TPL).</i> | |

G

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| get_id_title() | 57 |
| <i>returns the id and the title of the current node</i> | |
| get_node_content() | 61 |
| <i>Builds the content of the node by processing content.xml with xml2mode.xsl, and assigns it to a template variable.</i> | |
| get_short_xml_line() | 64 |
| <i>extract from the xml line a part around the error</i> | |
| get_node_references() | 68 |
| <i>Find the nodes that reference this one and return a list of them as a string (having a line for each node). The return value is empty string ("") if there are no referencies.</i> | |
| get_state_filename() | 40 |
| <i>Returns the file state.txt of the current node.</i> | |
| get_node_state() | 40 |
| <i>Reads the file state.txt of the current node and returns</i> | |
| GRAPHICS | 28 |

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| get_arr_languages() | 29 |
| <i>This function returns an associated array of language codes and language details (like language name, codeset, etc.).</i> | |
| get_arr_navigation() | 40 |
| <i>Reads from cache 'navigation.txt' for the given node and returns an associative array with the template variables <code>this_path</code>, <code>next_path</code>, <code>prev_path</code>, <code>up_path</code>, <code>this_title</code>, <code>next_title</code>, <code>prev_title</code>, <code>up_title</code>.</i> | |
| get_date_str() | 40 |
| <i>Converts the given time stamp to a date string.</i> | |
| get_user_data() | 3 |

H

| | |
|------------------------------------------------------------------------------------|----|
| help::on_select() | 33 |
| <i>this event is called by xref.php</i> | |
| html_to_xml() | 57 |
| help::get_langs() | 33 |
| <i>Returns an array of languages (en,fr,it,al) in which the book is available.</i> | |
| help::get_book_title() | 32 |
| help::afterParse() | 32 |
| help | 32 |
| <i>Displays a section of docbookwiki_guide.</i> | |

I

| | |
|------------------------------------------------------|----|
| Info::parse_copyright() | 77 |
| Info::parse_dummy() | 77 |
| Info::parse_author() | 77 |
| Info::parse_abstract() | 76 |
| Info::parse() | 76 |
| <i>Extract the various parts of the info section</i> | |
| Info::parse_keywords() | 77 |
| Info::render_abstract() | 77 |
| Info::to_xml() | 77 |
| <i>convert the info data to xml</i> | |
| Info::render_keywords() | 77 |
| Info::render_copyright() | 77 |
| Info::render_authors() | 77 |
| Info::\$keywords | 76 |
| Info::\$copyright | 76 |
| import_doc::on_import() | 7 |
| import_doc::on_refresh() | 7 |
| <i>refresh the list after uploading a file</i> | |
| import_doc::on_delete() | 7 |
| <i>delete an uploaded file</i> | |
| import_doc::onRender() | 7 |
| import_doc::add_uploaded_files_rs() | 7 |
| <i>create and add the recordset 'uploaded_files'</i> | |
| import_doc.php | 31 |

| | |
|-------------------------------------------------------------------------|----|
| is_locked() | 40 |
| <i>Returns true if the node is locked and the lock has not expired.</i> | |
| Info::\$authors | 76 |
| Info::\$abstract | 76 |
| Info | 76 |
| <i>Handle bookinfo or articleinfo elements.</i> | |
| is_wellformed() | 57 |
| <i>Check that the given XML string is well formed.</i> | |
| import_doc | 6 |
| <i>Import a new book or article in the system.</i> | |

L

| | |
|-------------------------------------------------------------------------------------------------|----|
| Listing::\$indent | 78 |
| Listing::\$bullet | 78 |
| Listing | 77 |
| lock_index() | 68 |
| Listing::to_html() | 78 |
| Listing::to_xml() | 78 |
| ListItem::to_xml() | 79 |
| ListItem::to_html() | 79 |
| ListItem | 78 |
| latex_to_xml() | 57 |
| langs::onRender() | 46 |
| languages::onParse() | 33 |
| languages::init() | 33 |
| languages | 33 |
| <i>Languages of the application interface.</i> | |
| languages::onRender() | 33 |
| languages::on_select() | 33 |
| langs | 46 |
| <i>A list of languages in which the book is available.</i> | |
| langs.php | 43 |
| locked_by_somebody() | 41 |
| <i>Returns true and displays a notification message, if the node is locked by another user.</i> | |
| languages.php | 30 |

M

| | |
|------------------------------------------------------------------------------------|----|
| main::init() | 35 |
| main::onRender() | 35 |
| main::get_langs() | 34 |
| <i>Returns an array of languages (en,fr,it,al) in which the book is available.</i> | |
| main::get_book_title() | 34 |
| main::check_book_id() | 34 |
| <i>Checks whether the given book_id exists or not.</i> | |
| main::on_approve() | 35 |
| main::on_book() | 35 |
| mediaobject() | 87 |

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <i>Returns a <mediobject> element</i> | |
| main::on_xref() | 35 |
| <i>this event is called by xref.php</i> | |
| main::on_search() | 35 |
| main::on_docbook() | 35 |
| main | 34 |
| <i>The starting point for constructing the web page.</i> | |
| MENU | 28 |
| modified_nodes::add_is_admin() | 20 |
| <i>Add in the recordset all the modified nodes in the books for which the user is admin.</i> | |
| modified_nodes::get_arr_books_accr() | 20 |
| <i>Return an array with the books where the user has some access rights and is not admin.</i> | |
| modified_nodes::add_has_accr() | 20 |
| <i>Find the modified nodes that can be approved by user and add them in the recordset.</i> | |
| modified_nodes | 19 |
| <i>Displays a list of the modified nodes that can be approved by the user.</i> | |
| modified_nodes.php | 18 |
| modified_nodes::get_arr_books_admin() | 20 |
| <i>return an array of the books for which the user is admin</i> | |
| modified_nodes::get_title() | 20 |
| modified_nodes::onRender() | 22 |
| <i>First find the books where the user has some access rights.</i> | |
| modified_nodes::node_match() | 21 |
| <i>Returns true if one of the node_path expressions in the list matches the given node path, otherwise returns false.</i> | |
| modified_nodes::lang_match() | 21 |
| <i>Returns true if one of the language ids in the list matches the given language, otherwise returns false. The lang_list is a comma separated list of languages, or 'ALL'.</i> | |
| modified_nodes::has_approve_right() | 21 |
| <i>Returns true if the user has approve right on the given node.</i> | |
| menu_items.php | 2 |
| <i>The \$menu_items array contains the items of the tabs1.</i> | |

N

| | |
|---------------------------------------------------------------------------------------------------------------------------------------|----|
| NULL_VALUE | 26 |
| <i>This constant is the value returned by the framework for a DB variable that has a NULL value. It can be "", "NULL", NULL, etc.</i> | |

P

| | |
|-------------------------------------------|----|
| package.WikiConverter.php | 74 |
| Paragraph | 79 |
| Paragraph::to_html() | 79 |
| Paragraph::to_xml() | 79 |
| process_index_node() | 68 |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------|----|
| <i>Applies the given transformer to index.xml, passing the node as a parameter, and returns the output.</i> | |
| putback_comments() | 62 |
| <i>replace <comment>x</comment> by the corresponding <!--...--></i> | |
| popup_window() | 60 |
| process_content.php | 61 |
| <i>This file contains functions about processing the content of a node that are used by the edit_content webbox and by other modules.</i> | |
| putback_cdata() | 62 |
| <i>replace <cdata>x</cdata> by the corresponding <![CDATA[...]]></i> | |
| process_index() | 41 |
| <i>applies the given transformer to index.xml</i> | |

R

| | |
|--------------------------------------------------------------------------|----|
| revisions::on_set_date() | 55 |
| revisions::on_save() | 55 |
| revisions::on_hide() | 54 |
| revisions::on_set_rev() | 55 |
| revisions::on_set_tag() | 55 |
| regex_replace() | 87 |
| remove_from_modified_nodes() | 68 |
| <i>Remove the node path from the file 'modified_nodes.txt'</i> | |
| revisions::on_show() | 55 |
| revisions::onRender() | 54 |
| revisions::init() | 54 |
| revisions::add_rs_revs() | 54 |
| <i>Add to webpage a recordset of revs.</i> | |
| revisions::add_rev_diff() | 54 |
| <i>add {{rev_diff}}, the difference</i> | |
| revisions::add_file_revision() | 54 |
| <i>add the variable {{file_revision}}</i> | |
| revisions::add_rs_tags() | 54 |
| <i>Add to webpage a recordset of tags.</i> | |
| revisions::clear() | 54 |
| revisions::get_svn_url() | 54 |
| <i>return the URL of the SVN repository: file://path/book_id/Ing</i> | |
| revisions::get_revision() | 54 |
| <i>return the content.xml at the selected revision (or date, or tag)</i> | |
| revisions | 54 |
| <i>Handle (view, select) the previous revisions of the book.</i> | |

S

| | |
|-----------------------------------------|----|
| search::on_new_search() | 36 |
| search::on_help() | 36 |
| search::onRender() | 36 |
| search::onParse() | 36 |
| search::on_next() | 36 |
| set_node_lock() | 41 |
| <i>Lock or unlock the current node.</i> | |

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| strip_comments() | 62 |
| <i>replace <!--...--> by <comment>x</comment></i> | |
| strip_cdata() | 62 |
| <i>replace <![CDATA[...]]> by <cdata>x</cdata></i> | |
| set_node_status() | 42 |
| <i>Set the status of the current node to the given new status.</i> | |
| search::init() | 36 |
| search::get_results_rs() | 36 |
| <i>Returns a recordset of search results.</i> | |
| SVN | 28 |
| SCRIPTS | 28 |
| SHOW_EXTERNAL_LINK | 25 |
| <i>If this constant is true, the framework outputs the external link that can be used to jump right to the current page of the application from any external page.</i> | |
| SHOW_ERROR_MESSAGES | 25 |
| <i>if this constant is false than the framework suppresses</i> | |
| search | 35 |
| <i>Search in the book(s) and display the results.</i> | |
| search::get_book_list() | 36 |
| <i>Read the file books/book_list and return the associative array \$arr_books, with keys 'book_id:lng' and with values 'book_title'.</i> | |
| search::get_query() | 36 |
| search::get_page_vars() | 36 |
| search::get_nr_of_recs() | 36 |
| save_user_data() | 3 |

T

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------|----|
| Tpl::\$id | 81 |
| Tpl::\$type | 81 |
| Tpl::get_subtemplates() | 81 |
| <i>Returns an array of ids of the templates that are referenced from the contents of this template.</i> | |
| Tpl::get_tpl_id() | 81 |
| <i>If the given line is a reference to another template, returns the id of the referenced template, otherwise return empty string "".</i> | |
| Tpl::\$contents | 81 |
| Tpl | 80 |
| Text | 80 |
| Text::line_to_html() | 80 |
| Text::line_to_xml() | 80 |
| Tpl::line_to_html() | 81 |
| Tpl::line_to_xml() | 82 |
| tabs1::add_items_list() | 91 |
| tabs1::init() | 91 |
| tabs1::onRender() | 91 |
| tabs1::on_select() | 91 |
| tabs1 | 91 |
| tabs1_items.php | 90 |
| <i>The \$menu_items array contains the items of the tabs1.</i> | |

| | |
|---------------------------------------------------------------------------------------------------------------|----|
| Tpl::to_html() | 82 |
| Tpl::to_html_table() | 82 |
| <i>Debug function.</i> | |
| Tpl::to_xml() | 82 |
| transform_index() | 68 |
| <i>Applies a transformation to index.xml and then writes the result back to index.xml.</i> | |
| text_to_xml() | 57 |
| tags::get_head_revision() | 49 |
| tags::get_revision_date() | 50 |
| tags::get_revision_nr() | 50 |
| tags::get_svn_url() | 50 |
| <i>return the URL of the SVN repository: file://path/book_id/Ing</i> | |
| tags::get_current_tag() | 49 |
| tags::get_book_dir() | 49 |
| <i>return the directory of the book: content/books/xml/book_id/Ing/</i> | |
| tags.php | 48 |
| tags | 49 |
| <i>Handle the tags of a book (create new tag, list existing tags, fix book to a tag, etc.)</i> | |
| tags::add_rs_tags() | 49 |
| <i>Add to webpage a recordset of tags.</i> | |
| tags::is_fixed() | 50 |
| tags::onRender() | 50 |
| tags::set_revision() | 50 |
| tags::write_fixed_file() | 51 |
| <i>Write the file 'fixed' which prevents the update of the latest changes of the book in the public copy.</i> | |
| texi_to_xml() | 57 |
| tags::on_set_tag() | 50 |
| tags::on_set_rev() | 50 |
| tags::on_add_tag() | 50 |
| tags::on_del_tag() | 50 |
| tags::on_set_date() | 50 |
| TPL | 28 |

U

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| userList::on_select() | 17 |
| userList::on_refresh() | 17 |
| userList::on_next() | 16 |
| userList::selectFirst() | 17 |
| <i>set currentUser as the first user in the list</i> | |
| user_data.php | 19 |
| user_data::add_rs_books_admin() | 22 |
| <i>Add to webPage a recordset of the books for which the user is admin.</i> | |
| user_data | 22 |
| <i>The user can change his data: name, email and password; he can also see which books he can manage, his edit rights and a list of nodes that are modified and he can approve.</i> | |
| userList::on_delete() | 16 |
| userList::on_add_new() | 16 |

| | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| userList::add_users_rs() | 16 |
| <i>Constructs and adds to the webpage the recordset 'users'.</i> | |
| userList | 16 |
| <i>userList displays the list of system users.</i> | |
| userList::get_nr_of_recs() | 16 |
| userList::get_page_vars() | 16 |
| userList::onRender() | 16 |
| userList::init() | 16 |
| user_data::onParse() | 22 |
| user_data::onRender() | 22 |
| update_cache_files_all_langs() | 69 |
| update_cache_files() | 69 |
| update_navigation() | 69 |
| <i>Update the file navigation.txt for the given node and language.</i> | |
| update_navigation_all_langs() | 69 |
| <i>Update the files navigation.txt for all languages, for the given node.</i> | |
| update_subnodes_html_all_langs() | 70 |
| <i>Updates the cache files subnodes.html for the given node, for all its ancestors, and for all the languages.</i> | |
| update_subnodes_html() | 70 |
| <i>Updates the cache files 'subnodes.html' for the given language, for the given node and all its ancestors, by applying 'update-subnodes.xsl' to index.xml.</i> | |
| updatenavig_moveup() | 69 |
| updatenavig_movedown() | 69 |
| user_data::on_save() | 22 |
| user_data::on_change_passwd() | 22 |
| USES_DB | 26 |
| <i>if this constant is true, the framework will load the DB component</i> | |
| upload_files() | 31 |
| <i>Upload the files 'xml_file' and 'media_files'.</i> | |
| unlock_index() | 69 |
| update_cache() | 42 |
| <i>Update the cache file content.html from content.xml</i> | |
| userFilter::onParse() | 16 |
| userFilter::init() | 16 |
| uList::add_users_rs() | 9 |
| <i>Constructs and adds to the webpage the recordset 'users'.</i> | |
| uList::add_all_users_rs() | 9 |
| <i>Constructs and adds to the webpage the recordset 'all_users' with all the users of the site.</i> | |
| uList | 9 |
| <i>The list of the users(editors) that can edit a book.</i> | |
| uList::get_nr_of_recs() | 9 |
| uList::get_page_vars() | 10 |
| uList::onRender() | 10 |
| uList::init() | 10 |
| uEdit::on_save() | 9 |
| <i>save the changes</i> | |
| uEdit::onRender() | 9 |
| uEdit::add_edit_rights() | 8 |
| uEdit | 8 |
| <i>Admin the edit rights of a user(editor) of a book.</i> | |

| | |
|-----------------------------------------------------------------------------------------------------------------|----|
| uEdit::add_rs_langs() | 8 |
| uEdit::add_rs_nodes() | 8 |
| uEdit::get_langs() | 9 |
| <i>Returns an array of languages (en,fr,it,al) in which the book is available.</i> | |
| uEdit::add_user_data() | 8 |
| <i>get the user data and add them to the template</i> | |
| uList::on_add_user() | 10 |
| uList::on_delete() | 10 |
| userEdit::on_add() | 15 |
| <i>add the new user in database</i> | |
| userEdit::onRender() | 14 |
| userEdit::on_delete() | 15 |
| <i>delete the current user</i> | |
| userEdit::on_save() | 15 |
| <i>save the changes</i> | |
| userFilter::buildFilter() | 16 |
| userFilter | 15 |
| <i>userFilter is used to select the users that are displayed in the list.</i> | |
| userEdit::onParse() | 14 |
| userEdit::init() | 14 |
| uList::on_select() | 10 |
| uList::on_next() | 10 |
| uList::selectFirst() | 10 |
| <i>set currentUser as the first user in the list</i> | |
| userEdit.php | 14 |
| userEdit::\$user_data | 14 |
| userEdit | 14 |
| <i>userEdit modifies the details of a user (username, password, name, email and which books can admin).</i> | |
| user_data.php | 3 |

V

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| validate.php | 64 |
| validate_xml() | 64 |
| view::setup_webnotes() | 46 |
| <i>set proper values to the parameters of the 'webnotes' component</i> | |
| view::onParse() | 46 |
| view | 46 |
| <i>Display the cache (html) file for the current node of the book.</i> | |
| view::init() | 46 |
| VAR_NOT_FOUND | 26 |
| <i>This constant sets the format of the error message that is displayed when a {{variable}} is not found. 'var_name' is replaced by the actual variable name. Examples: "'var_name' is not defined", "", "undefined", etc. It cannot contain "{{var_name}}" inside.</i> | |

W

| | |
|-------------------------------------------|----|
| WikiParser::parse_block() | 85 |
|-------------------------------------------|----|

| | |
|-------------------------------------------------------------------------|----|
| WikiParser::parse_file() | 85 |
| WikiParser::parse_info() | 85 |
| WikiParser::nonempty_line() | 84 |
| WikiParser::next_line() | 84 |
| <i>move the index to the nex line</i> | |
| WikiParser::listitem_start() | 84 |
| WikiParser::list_end() | 84 |
| WikiParser::list_start() | 84 |
| <i>return true if the current line starts a new list</i> | |
| WikiParser::parse_lines() | 85 |
| WikiParser::parse_list() | 85 |
| WikiParser::template_list() | 85 |
| <i>(debug) Outputs the data of each template in \$tpl_collection.</i> | |
| WikiParser::to_tree() | 85 |
| <i>Returns the structure of loaded templates as a tree.</i> | |
| WikiParser::tpl_to_tree() | 86 |
| <i>(debug) Returns the structure of the loaded templates as a tree.</i> | |
| WikiParser::parse_text() | 85 |
| <i>parse the text wiki</i> | |
| WikiParser::parse_string() | 85 |
| WikiParser::parse_listitem() | 85 |
| WikiParser::parse_paragraph() | 85 |
| WikiParser::listitem_end() | 84 |
| WikiParser::link_template() | 84 |
| <i>add a link from the parent template to the child template</i> | |
| WikiParser | 82 |
| WikiParser::\$arr_lines | 82 |
| <i>lines that will be parsed and their index</i> | |
| WikiParser::\$idx | 83 |
| write_node_state() | 42 |
| WS_CACHE | 28 |
| WOBJ | 28 |
| WORKSPACE | 28 |
| WS_BOOKS | 28 |
| WikiParser::append_line() | 83 |
| <i>append the current line to the given template</i> | |
| WikiParser::block_end() | 83 |
| WikiParser::get_current_line() | 83 |
| <i>get the current line that is being parsed</i> | |
| WikiParser::get_indentation() | 83 |
| <i>return the indentaion of the given mark</i> | |
| WikiParser::get_listitem_mark() | 84 |
| WikiParser::get_bullet() | 83 |
| <i>return the bullet of the given mark</i> | |
| WikiParser::end_of_lines() | 83 |
| <i>return true if the index is at the end of the lines</i> | |
| WikiParser::block_start() | 83 |
| WikiParser::empty_line() | 83 |
| WEBAPP_PATH | 28 |

| | |
|-------------------------------|----|
| xml_to_xml() | 58 |
| xml_to_text() | 56 |
| xml_to_html() | 56 |
| XSLT | 28 |