

ResourceSpace Function Reference and Technical Notes

Application Overview

The application is coded in a procedural fashion for simplicity and for maximum portability. My approach is to make the solution as simple as possible. The PHP files in the root directory are used like templates with presentation logic only (loops, conditions, user event handling). All application logic is contained in functions within the included files in the 'include' directory. These functions are documented below.

Several include files are used, as functions are grouped into specific functions, such as search, editing, collections etc. A page will only include files it needs rather than including all files.

Function List

add_collection (\$user, \$collection)

Add a collection to a user's 'My Collections'

[collections_functions.php](#)

add_keyword_mappings (\$ref, \$string, \$resource_type_field)

For each instance of a keyword in \$string, add a keyword->resource mapping. Create keywords that do not yet exist. Increase the hit count of each keyword that matches. Store the position and field the string was entered against for advanced searching.

[resource_functions.php](#)

add_resource_to_collection (\$resource, \$collection)

[collections_functions.php](#)

add_saved_search (\$collection)

[collections_functions.php](#)

average_length (\$array)

Returns the average length of the strings in an array

[general.php](#)

checkperm (\$perm)

Check that the user has the \$perm permission

[db.php](#)

copy_resource (\$from, \$resource_type=-1)

Create a new resource, copying all data from the resource with reference \$from. Note this copies only the data and not any attached file. It's very unlikely the same file would be in the system twice, however users may want to clone an existing resource To avoid reentering data if the resource is very similar. If \$resource_type is specified then the resource type for the new resource will be set to \$resource_type Rather than simply copied from the \$from resource.

[resource_functions.php](#)

create_collection (\$userid, \$name, \$allowchanges=0, \$cant_delete=0)

Creates a new collection and returns the reference

[collections_functions.php](#)

create_previews (\$ref, \$thumbonly=false)

For resource \$ref, (re)create the various preview sizes listed in the table preview_sizes Only create previews where the target size IS LESS THAN OR EQUAL TO the source size. Set thumbonly=true to (re)generate thumbnails only.

[image_processing.php](#)

create_resource (\$resource_type)

Create a new resource.

[resource_functions.php](#)

daily_stat (\$activity_type, \$object_ref)

Update the daily statistics after a loggable event. The daily_stat table contains a counter for each 'activity type' (i.e. download) for each object (i.e. resource) Per day.

[db.php](#)

delete_collection (\$ref)

Deletes the collection with reference \$ref

[collections_functions.php](#)

delete_resource (\$ref)

Delete the resource, all related entries in tables and all files on disk

[resource_functions.php](#)

do_report (\$ref, \$from_y, \$from_m, \$from_d, \$to_y, \$to_m, \$to_d)

Run report with id \$ref for the date range specified. Returns a result array.

[reporting_functions.php](#)

do_search (\$search, \$restypes="", \$order_by="relevance", \$archive=0, \$fetchrows=-1)

Takes a search string \$search, as provided by the user, and returns a results set Of matching resources. If there are no matches, instead returns an array of suggested searches. \$restypes is optionally used to specify which resource types to search.

[search_functions.php](#)

email_collection (\$collection, \$collectionname, \$fromusername, \$userlist, \$message)

Attempt to resolve all users in the string \$userlist to user references. Add \$collection to these user's 'My Collections' page Send them an e-mail linking to this collection

[collections_functions.php](#)

email_reminder (\$email)

[general.php](#)

email_resource (\$resource, \$resourcename, \$fromusername, \$userlist, \$message)

Attempt to resolve all users in the string \$userlist to user references. Add \$collection to these user's 'My Collections' page Send them an e-mail linking to this collection

[resource_functions.php](#)

email_resource_request (\$ref, \$details)

E-mails a resource request (posted) to the team

[general.php](#)

email_user_request ()

E-mails the submitted user request form to the team.

[general.php](#)

errorhandler (\$errno, \$errstr, \$errfile, \$errline)

[db.php](#)

escape_check (\$text) #only escape a string if we need to, to prevent escaping an already escaped string

[db.php](#)

extract_mean_colour (\$image, \$ref)

For image \$image, calculate the mean colour and update this to the image_red, image_green, image_blue tables In the resources table. Also - we insert the height and width of the thumbnail at this stage as all information is available and we Are already performing an update on the resource record.

[image_processing.php](#)

formatfilesize (\$bytes)

Return a human-readable string representing \$bytes in either KB or MB.

[general.php](#)

get_active_users ()

Returns a list of active users, i.e. users still logged on with a last-active time within the last 2 hours.

[general.php](#)

get_advanced_search_fields ()

Returns a list of fields suitable for advanced searching.

[search_functions.php](#)

get_all_site_text (\$find="")

Returns a list of all available editable site text (content). If \$find is specified a search is performed across page, name and text fields.

[general.php](#)

get_collection (\$ref)

Returns all data for collection \$ref

[collections_functions.php](#)

get_data_by_field (\$resource, \$field)

Return the resource data for field \$field in resource \$resource

[general.php](#)

get_field_options (\$ref)

For the field with reference \$ref, return a sorted array of options.

[general.php](#)

get_image_sizes (\$ref, \$internal=false)

Returns a table of available image sizes for resource \$ref. The original image file assumes the name of the 'nearest size (up)' in the table

[general.php](#)

get_max_resource_ref ()

Returns the highest resource reference in use.

[resource_functions.php](#)

get_related_resources (\$ref)

Return an array of resource references that are related to resource \$ref

[general.php](#)

get_reports ()

Returns all reports in a result array.

[reporting_functions.php](#)

get_research_request (\$ref)

[research_functions.php](#)

get_research_request_collection (\$ref)

[research_functions.php](#)

get_research_requests (\$find="")

[research_functions.php](#)

get_resource_data (\$ref)

Returns basic resource data (from the resource table alone) for resource \$ref. For 'dynamic' field data, see

get_resource_field_data

[general.php](#)

get_resource_field_data (\$ref)

Returns field data and field properties (resource_type_field and resource_data tables) For this resource, for display in an edit / view form.

[general.php](#)

get_resource_field_data_batch (\$refs)

Returns field data and field properties (resource_type_field and resource_data tables) For all the resource references in the array \$refs. This will use a single SQL query and is therefore a much more efficient way of gathering Resource data for a list of resources (e.g. search result display for a page of resources).

[general.php](#)

get_resource_path (\$ref, \$size, \$generate, \$extension="jpg")

Returns the correct path to resource \$ref of size \$size (\$size==empty string is original resource) If one or more of the folders do not exist, and \$generate=true, then they are generated

[general.php](#)

get_resource_ref_range (\$lower, \$higher)

Returns an array of resource references in the range \$lower to \$upper.

[resource_functions.php](#)

get_resource_top_keywords (\$resource, \$count)

Return the top \$count keywords (by hitcount) used by \$resource. This is for the 'Find Similar' search. Keywords that are too short or too long, or contain numbers are dropped - they are probably not as meaningful in The contexts of this search (consider being offered "12" or "OKB-34" as an option?)

[general.php](#)

get_resource_types ()

Returns a list of resource types.

[general.php](#)

get_saved_searches (\$collection)

[collections_functions.php](#)

get_section_list (\$page)

[db.php](#)

get_site_text (\$ref)

Returns a specific site text entry.

[general.php](#)

get_stats_activity_types ()

Returns a list of activity types for which we have stats data (Search, User Session etc.)

[general.php](#)

get_stats_years ()

Returns a list of years for which we have statistics.

[general.php](#)

get_theme_headers ()

Return a list of theme headers, i.e. theme categories

[collections_functions.php](#)

get_themes (\$header)

For a given header, return a list of themes

[collections_functions.php](#)

get_user (\$ref)

Return a user's credentials.

[general.php](#)

get_user_collections (\$user)

Returns a list of user collections. Additionally, if the user hasn't got any collections, then create a default 'My Collection'. \$list=sql_query("select c.,u.username,count(r.resource) count from user u,collection c left outer join collection_resource r on c.ref=r.collection where u.ref=c.user and c.user='\$user' group by c.ref union select c.*,u.username,count(r.resource) count from user_collection uc left join collection c on uc.collection=c.ref left outer join collection_resource r on c.ref=r.collection left join user u on c.user=u.ref where uc.user='\$user' group by c.ref order by created;");*

[collections_functions.php](#)

get_usergroups ()

Returns a list of user groups.

[general.php](#)

get_users (\$group=0, \$find="")

Returns a user list. Group or search team is optional.

[general.php](#)

getuid ()

Generate a unique ID

[db.php](#)

getval (\$val, \$default)

Return a value from get/post or a default if neither set

[db.php](#)

getvalescaped (\$val, \$default)

Return a value from get/post, escaped and SQL-safe

[db.php](#)

new_user (\$newuser)

Create a new user with username \$newuser. Returns the created user reference.

[general.php](#)

newlines (\$text)

Replace escaped newlines with real newlines.

[general.php](#)

nicedate (\$date, \$time=false, \$wordy=false)

Format a MySQL ISO date in the UK style

[db.php](#)

pagename ()

[db.php](#)

redirect (\$url)

Redirect to a relative URL

[db.php](#)

refresh_collection_frame ()

[collections_functions.php](#)

remove_collection (\$user, \$collection)

Remove someone else's collection from a user's My Collections

[collections_functions.php](#)

remove_keyword_mappings (\$ref, \$string, \$resource_type_field)

Removes one instance of each keyword->resource mapping for each occurrence of that Keyword in \$string. This is used to remove keyword mappings when a field has changed. We also decrease the hit count for each keyword.

[resource_functions.php](#)

remove_resource_from_collection (\$resource, \$collection)

[collections_functions.php](#)

remove_saved_search (\$collection, \$search)

[collections_functions.php](#)

resolve_keyword (\$keyword)

Returns the keyword reference for \$keyword, or false if no such keyword exists.

[general.php](#)

resolve_soundex (\$keyword)

Returns the most commonly used keyword that sounds like \$keyword, or failing a soundex match, The most commonly used keyword that starts with the same few letters.

[search_functions.php](#)

resolve_user_agent (\$agent)

[db.php](#)

rotate_preview_images (\$ref)

Rotate all preview images 90 degrees anti-clockwise This option is provided on the edit screen because images may be stored the wrong way up as this is how they came from the camera, and rotating the master images would potentially reduce the quality. So we rotate the previews for convenience

[image_processing.php](#)

save_collection (\$ref)

Update collection with submitted form data

[collections_functions.php](#)

save_research_request (\$ref)

Save

[research_functions.php](#)

save_resource_data (\$ref)

Save all submitted data for resource \$ref. Also re-index all keywords from indexable fields.

[resource_functions.php](#)

save_site_text (\$ref)

Saves the submitted site text changes to the database.

[general.php](#)

save_user (\$ref)

Save user details, data is taken from the submitted form.

[general.php](#)

search_public_collections (\$search)
[collections_functions.php](#)

send_research_request ()
Insert a search request into the requests table.
[research_functions.php](#)

set_research_collection (\$research, \$collection)
[research_functions.php](#)

set_user_collection (\$user, \$collection)
[collections_functions.php](#)

split_keywords (\$search, \$index=false)
Takes \$search and returns an array of individual keywords.
[general.php](#)

sql_array (\$query)
Like sql_value() but returns an array of all values found. The value returned must have the column name aliased to 'value'
[db.php](#)

sql_insert_id ()
Return last inserted ID (abstraction)
[db.php](#)

sql_query (\$sql, \$cache=false, \$fetchrows=-1)
Sql_query(sql) - execute a query and return the results as an array. Database functions are wrapped in this way so supporting a database server other than MySQL is Easier. \$cache is not used at this time - it was intended for disk based results caching which may be added in the future. If \$fetchrows is set we don't have to loop through all the returned rows. We Just fetch \$fetchrows row but pad the array to the full result set size with empty values. This has been added retroactively to support large result sets, yet a pager can work as if a full Result set has been returned as an array (as it was working previously).
[db.php](#)

sql_value (\$query, \$default)
Return a single value from a database query, or the default if no rows The value returned must have the column name aliased to 'value'
[db.php](#)

string_similar (\$string1, \$string2)
Returns an integer score based on how similar the two strings are. This was used when importing data for "fuzzy" keyword/option matching.
[general.php](#)

suggest_refinement (\$refs, \$search)
Given an array of resource references (\$refs) and the original Search query (\$search), produce a list of suggested search refinements to Reduce the result set intelligently.
[search_functions.php](#)

text (\$name)
Returns site text with name \$name, or failing that returns dummy text.
[db.php](#)

tidy_trim (\$text, \$length)
Trims \$text to \$length if necessary. Tries to trim at a space if possible. Adds three full stops If trimmed...
[general.php](#)

tidylist (\$list)
Takes a value as returned from a check-list field type and reformats to be more display-friendly. Check-list fields have a leading comma.
[general.php](#)

trim_array (\$array)
Removes whitespace from the beginning/end of all elements in an array
[general.php](#)

trim_spaces (\$text)

Replace multiple spaces with a single space

[general.php](#)

update_resource_keyword_hitcount (\$resource, \$search)

For the specified \$resource, increment the hitcount for each matching keyword in \$search

[general.php](#)

upload_file (\$ref)

Process file upload for resource \$ref

[image_processing.php](#)

Database Schema

Tables_in_oasis
collection
collection_resource
collection_savedsearch
daily_stat
keyword
preview_size
report
research_request
resource
resource_data
resource_keyword
resource_related
resource_type
resource_type_field
site_text
user
user_collection
usergroup

Table collection						
Field	Type	Null	Key	Default	Extra	
ref	int(11)	NO	PRI	NULL	auto_increment	
name	varchar(100)	YES		NULL		
user	int(11)	YES		NULL		
created	datetime	YES		NULL		
public	int(11)	NO		0		
theme	varchar(100)	YES		NULL		
allow_changes	int(11)	YES		0		
cant_delete	int(11)	NO		0		

Table collection_resource						
Field	Type	Null	Key	Default	Extra	
collection	int(11)	YES	MUL	NULL		
resource	int(11)	YES		NULL		
date_added	timestamp	YES		CURRENT_TIMESTAMP		

Table collection_savedsearch						
Field	Type	Null	Key	Default	Extra	
collection	int(11)	YES		NULL		
search	text	YES		NULL		
restypes	text	YES		NULL		
archive	int(11)	YES		NULL		
ref	int(11)	NO	PRI	NULL	auto_increment	
created	timestamp	YES		CURRENT_TIMESTAMP		

Table daily_stat						
Field	Type	Null	Key	Default	Extra	
year	int(11)	YES	MUL	NULL		
month	int(11)	YES		NULL		
day	int(11)	YES		NULL		
activity_type	varchar(50)	YES		NULL		
object_ref	int(11)	YES		NULL		
count	int(11)	YES		0		

Table keyword						
---------------	--	--	--	--	--	--

Field	Type	Null	Key	Default	Extra
ref	int(11)	NO	PRI	NULL	auto_increment
keyword	varchar(100)	YES	MUL	NULL	
soundex	varchar(10)	YES		NULL	
hit_count	int(11)	NO	MUL	0	

Table preview_size					
Field	Type	Null	Key	Default	Extra
id	char(3)	YES		NULL	
width	int(11)	YES		NULL	
height	int(11)	YES		NULL	
padtosize	int(11)	YES		0	
name	varchar(50)	YES		NULL	
internal	int(11)	YES		0	
allow_preview	int(11)	YES		0	

Table report					
Field	Type	Null	Key	Default	Extra
ref	int(11)	NO	PRI	NULL	auto_increment
name	varchar(100)	YES		NULL	
query	text	YES		NULL	

Table research_request					
Field	Type	Null	Key	Default	Extra
ref	int(11)	NO	PRI	NULL	auto_increment
name	text	YES		NULL	
description	text	YES		NULL	
deadline	datetime	YES		NULL	
contact	varchar(100)	YES		NULL	
finaluse	text	YES		NULL	
resource_types	varchar(50)	YES		NULL	
noresources	int(11)	YES		NULL	
shape	varchar(50)	YES		NULL	
created	datetime	YES		NULL	
user	int(11)	YES		NULL	
assigned_to	int(11)	YES		NULL	
status	int(11)	NO		0	
collection	int(11)	YES	MUL	NULL	

Table resource					
Field	Type	Null	Key	Default	Extra
ref	int(11)	NO	PRI	NULL	auto_increment
title	text	YES		NULL	
resource_type	int(11)	YES	MUL	NULL	
has_image	int(11)	YES		0	
hit_count	int(11)	NO	MUL	0	
creation_date	datetime	YES	MUL	NULL	
rating	int(11)	YES	MUL	NULL	
country	text	YES		NULL	
file_extension	varchar(10)	YES		NULL	
image_red	int(11)	YES		NULL	
image_green	int(11)	YES		NULL	
image_blue	int(11)	YES		NULL	
thumb_width	int(11)	YES		NULL	
thumb_height	int(11)	YES		NULL	
archive	int(11)	YES	MUL	0	
access	varchar(50)	NO	MUL		
temp_noteid	varchar(50)	YES	MUL	NULL	
temp_attempted_ftp	int(11)	NO		0	
temp_links	text	YES		NULL	

+-----+

Table resource_data						
Field	Type	Null	Key	Default	Extra	
resource	int(11)	YES	MUL	NULL		
resource_type_field	int(11)	YES	MUL	NULL		
value	text	YES		NULL		

Table resource_keyword						
Field	Type	Null	Key	Default	Extra	
resource	int(11)	NO	MUL			
keyword	int(11)	NO	MUL			
hit_count	int(11)	YES		0		
position	int(11)	YES		0		
resource_type_field	int(11)	YES	MUL	0		

Table resource_related						
Field	Type	Null	Key	Default	Extra	
resource	int(11)	NO	MUL			
related	int(11)	NO	MUL			

Table resource_type						
Field	Type	Null	Key	Default	Extra	
ref	int(11)	NO	PRI	NULL	auto_increment	
name	varchar(50)	YES		NULL		

Table resource_type_field						
Field	Type	Null	Key	Default	Extra	
ref	int(11)	NO	PRI	NULL	auto_increment	
name	varchar(50)	YES		NULL		
title	varchar(50)	YES		NULL		
type	int(11)	YES		NULL		
options	text	YES		NULL		
order_by	int(11)	NO		0		
keywords_index	int(11)	NO		0		
resource_type	int(11)	NO	MUL	0		
resource_column	varchar(50)	YES		NULL		
display_field	int(11)	YES		1		

Table site_text						
Field	Type	Null	Key	Default	Extra	
page	varchar(50)	YES		NULL		
name	varchar(50)	YES		NULL		
text	text	YES		NULL		
ref	int(11)	NO	PRI	NULL	auto_increment	

Table user						
Field	Type	Null	Key	Default	Extra	
ref	int(11)	NO	PRI	NULL	auto_increment	
username	varchar(50)	YES		NULL		
password	varchar(50)	YES		NULL		

fullname	varchar(100)	YES		NULL	
email	varchar(100)	YES		NULL	
usergroup	int(11)	YES		NULL	
last_active	datetime	YES		NULL	
logged_in	int(11)	YES		NULL	
last_browser	varchar(100)	YES		NULL	
last_ip	varchar(100)	YES		NULL	
current_collection	int(11)	YES		NULL	
accepted_terms	int(11)	NO		0	
account_expires	datetime	YES		NULL	

Table user_collection					
Field	Type	Null	Key	Default	Extra
user	int(11)	YES		NULL	
collection	int(11)	YES		NULL	

Table usergroup					
Field	Type	Null	Key	Default	Extra
ref	int(11)	NO	PRI	NULL	auto_increment
name	varchar(50)	YES		NULL	
permissions	text	YES		NULL	

User Permissions

A user is a member of a single group, and the group defines what they can do within the application. Users are managed within the team centre in the application, however groups management is considered more advanced and is only available on the 'System Setup' tree normally only available to administrators. Each group has a related permissions string which is simply a comma separated list of codes defining which activities the users can carry out. Those permissions are as follows:

Search

```
s      Can search for resources
v      Can view restricted (admin only) resources, also download 'confidential' resources
```

Resource creation

```
c      Can create resources / upload files
e?     Can edit resources in specific archive state, e.g. e0, e1, e2
       where the appended number is the archive state:
       0: Not archived
       1: Waiting to be archived
       2: Archived
       Typically the resources team can edit in states 0 and 1, whereas the archive team can edit
       in states 1 and 2.
i      Can manage archive resources (link appears in team centre)
```

Themes / Collections

```
h      Can publish themes, and edit all collections
```

Administration

```
a      Can access administration tree
t      Can see the team centre home
r      Can manage research requests
u      Can manage users
o      Can manage content
```