

Please take note that this work is still ongoing. The documentation should grow in a near future. 😊

Internals

In this chapter, we will provide extensive information on what Sympa looks like “under the hood”. We hope this documentation will be particularly helpful to developers.

Focus on the main Sympa modules

This chapter describes these modules (or a part of them):

- [src/mail.pm](#): low level of email sending;
- [src/List.pm](#): list processing and informations about structure and access to list configuration parameters;
- [src/sympa.pl](#): the main script, for messages and mail commands processing;
- [src/Commands.pm](#): mail commands processing;
- [wwsympa/wwsympa.fcgi](#): web interface;
- [src/report.pm](#): notification and error reports about requested services (email and web);
- [src/tools.pm](#): various tools;
- [src/Message.pm](#): message object used to encapsule a message received.
- [wwsympa/SympaSession.pm](#): session object use to manage user context with a single cookie and a session table
- [wwsympa/Auth.pm](#)  to be completed
- [wwsympa/ajax](#): ajax ;

A [graph of subroutines for messages process](#)

Sympa modules sorted by directory

This section presents all the files used in the Sympa engine. The files described are those contained in the sources, consequently:

1. you will not find the files generated at compilation time,
2. the files are not located in the same directory as you will find them in the compiled application.

Failing a better alternative, we present them as they appear in the `src` directory from the source distribution.

"./" : Sympas sources root directory

These scripts are run at installation time. They perform the operations necessary to the correct install and runtime of Sympa.

- [check_perl_modules.pl](#) : This script checks installed and required Perl modules. It also does the required installations.
- [important_changes.pl](#) : This script prints important changes in Sympa since last install. It is based on the NEWS ***** entries.
- [set_symlinks.pl](#) : This script sets symbolic links at installation time.
- [subst.pl](#) : This script replaces --VAR-- occurrences at installation time.

"./doc/"

-  [parse_tex.pl](#)

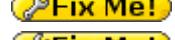
"./po/"

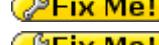
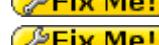
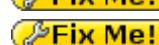
-  [check_locales.pl](#)
-  [xgettext.pl](#)

"./soap/"

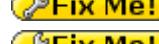
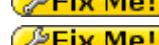
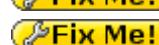
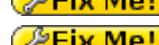
-  [sympa_soap_client.pl](#)
-  [sympasoap.pm](#)
-  [sympa_soap_server.fcgi](#)
-  [SympaTransport.pm](#)

"./src/"

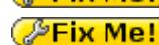
-  [admin.pm](#)
-  [alias_manager.pl](#)
-  [aliaswrapper.c](#)
-  [Archive.pm](#)
-  [bouncequeue.c](#)
-  [CAS.pm](#)
- [Commands.pm](#)
-  [Config_XML.pm](#)
-  [Conf.pm](#)
-  [Datasource.pm](#)
-  [Family.pm](#)
-  [familyqueue.c](#)
-  [Fetch.pm](#)
-  [Language.pm](#)
-  [Ldap.pm](#)
- [List.pm](#)
-  [Lock.pm](#)
-  [Log.pm](#)
- [mail.pm](#)
- [Message.pm](#)
-  [parser.pl](#)
-  [PlainDigest.pm](#)

-  queue.c
- report.pm
-  SQLSource.pm
- sympa.pl
-  sympa_wizard.pl
-  task_manager.pl
-  Task.pm
-  time_utils.pl
- tools.pl
-  tt2.pl
-  Upgrade.pm
-  Version.pm
-  virtualwrapper.c

"./src/etc/script/"

-  arc2webarc.pl
-  arcrepair.pl
-  crypt_passwd.pl
-  find_missing_messages.pl
-  init_comment.pl
-  ldap_alias_manager.pl
-  mod2html.pl
-  mysql_alias_manager.pl
-  p12topem.pl
-  testldap.pl
-  testlogs.pl
-  tpl2tt2.pl

"./wwsympa/"

-  archived.pl
-  Auth.pm
-  bounced.pl
-  bounce-lib.pl
-  cookielib.pm
-  Marc.pm
-  SharedDocument.pm
-  wwslib.pm
- wwsympa.fcgi
-  wwsympa_sudo_wrapper.pl

"./wwsympa/Marc/"

-  Search.pm

Database

Work in progress. All the tables are described already though.

This chapter describes the Sympa database.

subscriber_table

This table store subscription, subscription option etc..

Fields	Type	Properties	Usage
user_subscriber	varchar(100)	primary key	email of subscriber
list_subscriber	varchar(50)	primary key	list name of a subscription
robot_subscriber	varchar(80)	primary key	robot (domain) of the list
reception_subscriber	varchar(20)		reception format option of subscriber (digest, summary, etc)
suspend_subscriber	int(1)		boolean set to 1 if subscription is suspended
suspend_start_date_subscriber	int(11)		The date (epoch) when message reception is suspended
suspend_end_date_subscriber	int(11)		The date (epoch) when message reception should be restored
bounce_subscriber	varchar(35)		
bounce_score_subscriber	smallint(6)		
custom_attribute_subscriber	text		
bounce_address_subscriber	varchar(100)		
date_subscriber	datetime		date of subscription
update_subscriber	datetime		the latest date where subscription is confirmed by subscriber
comment_subscriber	varchar(150)		Free form name
number_messages_subscriber	int(5)		the number of message the subscriber sent
visibility_subscriber	varchar(20)		
topics_subscriber	varchar(200)		topic subscription specification
subscribed_subscriber	int(1)		boolean set to 1 if subscriber comes from ADD or SUB
included_subscriber	int(1)		boolean, set to 1 is subscriber comes from an external datasource. Note that included_subscriber and subscribed_subscriber can both value 1
include_sources_subscriber	varchar(50)		comma seperated list of datasource that contain this subscriber

user_table

The user_table is mainly used to manage login from web interface. A subscriber may not appear in the user_table if he never log through the web interface..

Fields	Type	Properties	Usage
email_user	varchar(100)	primary key	email user is the key
attributes_user	text		
data_user	text		
cookie_delay_user	int(11)		
lang_user	varchar(10)		user language preference
password_user	varchar(40)		password are stored as finger print
gecos_user	varchar(150)		
last_login_date_user	int(11)		date epoch from last login, printed in login result for security purpose
last_login_host_user	varchar(60)		host of last login, printed in login result for security purpose
wrong_login_count_user	int(11)		login attempt count, used to prevent brut force attack

spool_table

This table is created in version 6.3. It replace most of spools on file system for clustering purpose.

Fields	Type	Properties	Usage
messagekey_spool	bigint(20)	primary key	autoincrement key
spoolname_spool	enum('msg','auth','mod','digest','archive','bounce','subscribe','topic','bulk','validated','task')		the spool name
list_spool	varchar(50)		
robot_spool	varchar(80)		
priority_spool	varchar(2)		priority (list priority, owner priority etc)
date_spool	int(11)		the date a message is copied in spool table
message_spool	longtext		message as string b64 encoded
messagelock_spool	varchar(90)		a unique string for each process : \$\$@hostname
lockdate_spool	int(11)		the date a lock is set. Used in order detect old locks
message_status_spool	enum('ok','bad')		if problem when processed entries have bad status
message_diag_spool	text		the reason why a message is moved to bad
type_spool	varchar(15)		list, list-request,, sympa robot or other rcp
authkey_spool	varchar(33)		authentication key for email challenge
headerdate_spool	varchar(80)		the message header date

Fields	Type	Properties	Usage
create_list_if_needed_spool	int(1)		set to 1 if message is related to a dynamic list, set to 0 if list as been created or if list is static
subject_spool	varchar(110)		subject of the message stored to list spool content faster
sender_spool	varchar(110)		this info is stored to browse spool content faster
messageid_spool	varchar(300)		stored to list spool content faster
spam_status_spool	varchar(12)		spamstatus scenario result
size_spool	int(11)		info stored in order to browse spool content faster
task_date_spool	int(11)		date for a task
task_label_spool	varchar(20)		label for a task
task_model_spool	varchar(40)		model of related task
task_object_spool	varchar(50)		object of related task
dkim_privatekey_spool	varchar(1000)		DKIM parameter stored for bulk daemon because bulk ignore list parameters, private key to sign message
dkim_selector_spool	varchar(50)		DKIM parameter stored for bulk daemon because bulk ignore list parameters, DKIM selector to sign message
dkim_d_spool	varchar(50)		DKIM parameter stored for bulk daemon because bulk ignore list parameters, the d DKIM parameter
dkim_i_spool	varchar(100)		DKIM parameter stored for bulk daemon because bulk ignore list parameters, DKIM i signature parameter

Fields	Type	Properties	Usage
dkim_header_list_spool	varchar(500)		DKIM parameter stored for bulk daemon because bulk ignore list parameters, the list of headers included in signature

bulkmailer_table

storage of recipients with a ref to a message in spool_table. So a very simple process can distribute them.

Fields	Type	Properties	Usage
messagekey_bulkmailer	varchar(80)	primary key	A pointer to a message in spool_table. It must be a value of a line in table spool_table with same value as messagekey_spool
packetid_bulkmailer	varchar(33)	primary key	An id for the packet
messageid_bulkmailer	varchar(200)		The message Id
recipients_bulkmailer	text		the comma separated list of recipient email for this message
returnpath_bulkmailer	varchar(100)		the return path value that must be set when sending the message
robot_bulkmailer	varchar(80)		
listname_bulkmailer	varchar(50)		
verp_bulkmailer	int(1)		A boolean to specify if VERP is required, in this case return_path will be formatted using verp form
tracking_bulkmailer	enum('mdn','dsn')		Is DSN or MDM required when sending this message?
merge_bulkmailer	int(1)		Boolean, if true, the message is to be parsed as a TT2 template foreach recipient
priority_message_bulkmailer	smallint(10)		
priority_packet_bulkmailer	smallint(10)		
reception_date_bulkmailer	int(11)		The date where the message was received
delivery_date_bulkmailer	int(11)		The date the message was sent
lock_bulkmailer	varchar(30)		A lock. It is set as process-number @ hostname so multiple bulkmailer can handle this spool

exclusion_table

exclusion table is used in order to manage unsubscription for subscriber included from an external data source.

Fields	Type	Properties	Usage
user_exclusion	varchar(100)	primary key	
date_exclusion	int(11)		
list_exclusion	varchar(50)	primary key	

session_table

management of http session.

Fields	Type	Properties	Usage
id_session	varchar(30)	primary key	the identifier of the database record
start_date_session	int(11)		the date when the session was created
date_session	int(11)		date epoch of the last use of this session. It is used in order to expire old sessions
remote_addr_session	varchar(60)		The IP address of the computer from which the session was created
robot_session	varchar(80)		The virtual host in which the session was created
email_session	varchar(100)		the email associated to this session
hit_session	int(11)		the number of hit performed during this session. Used to detect crawlers
data_session	text		parameters attached to this session that don't have a dedicated column in the database

one_time_ticket_table

One time ticket are random value use for authentication challenge. A ticket is associated with a context which look like a session.

Fields	Type	Properties	Usage
data_one_time_ticket	varchar(200)		
status_one_time_ticket	varchar(60)		
remote_addr_one_time_ticket	varchar(60)		
ticket_one_time_ticket	varchar(30)	primary key	
date_one_time_ticket	int(11)		
robot_one_time_ticket	varchar(80)		
email_one_time_ticket	varchar(100)		

notification_table

used for message tracking feature. If the list is configured for tracking, outgoing messages include a delivery status notification request and optionnaly a return receipt request. When DSN MDN are received by Sympa, they are stored in this table in relation with the related list and message_id.

Fields	Type	Properties	Usage
date_notification	int(11)		
pk_notification	bigint(20)	primary key	Autoincrement key

Fields	Type	Properties	Usage
message_id_notification	varchar(100)		initial message-id. This feild is used to search DSN and MDN related to a particular message
recipient_notification	varchar(100)		email adresse of recipient for which a DSN or MDM was received
reception_option_notification	varchar(20)		The subscription option of the subscriber when the related message was sent to the list. Ussefull because some recipient may have option such as <i>digest</i> or <i>nomail</i>
status_notification	varchar(100)		Value of notification
arrival_date_notification	varchar(80)		reception date of latest DSN or MDM
type_notification	enum('DSN', 'MDN')		Type of the notification (DSN or MDM)
message_notification	longtext		The DSN or the MDN itself
list_notification	varchar(50)		The listname the messaage was issued for
robot_notification	varchar(80)		The robot the message is related to

logs_table

Each important event is stored in this table. List owners and listmaster can search entries in this table using web interface..

Fields	Type	Properties	Usage
id_logs	bigint(20)	primary key	Unique log's identifier
user_email_logs	varchar(100)		e-mail address of the message sender or email of identified web interface user (or soap user)
date_logs	int(11)		date when the action was executed
robot_logs	varchar(80)		name of the robot in which context the action was executed
list_logs	varchar(50)		name of the mailing-list in which context the action was executed
action_logs	varchar(50)		name of the Sympa subroutine which initiated the log
parameters_logs	varchar(100)		List of commas-separated parameters. The amount and type of parameters can differ from an action to another
target_email_logs	varchar(100)		e-mail address (if any) targeted by the message
msg_id_logs	varchar(255)		identifier of the message which triggered the action
status_logs	varchar(10)		exit status of the action. If it was an error, it is likely that the error_type_logs field will contain a description of this error
error_type_logs	varchar(150)		name of the error string - if any - issued by the subroutine
client_logs	varchar(100)		IP address of the client machine from which the message was sent
daemon_logs	varchar(10)		name of the Sympa daemon which ran the action

stat_table

Statistic item are store in this table, Sum average etc are stored in Stat_counter_table.

Fields	Type	Properties	Usage
daemon_stat	varchar(10)		
parameter_stat	varchar(50)		
list_stat	varchar(150)		
id_stat	bigint(20)	primary key	
operation_stat	varchar(50)		
email_stat	varchar(100)		
user_ip_stat	varchar(100)		
read_stat	tinyint(1)		
date_stat	int(11)		
robot_stat	varchar(80)		

stat_counter_table

Use in conjunction with stat_table for users statistics.

Fields	Type	Properties	Usage
list_counter	varchar(150)		
end_date_counter	int(11)		
robot_counter	varchar(80)		
id_counter	bigint(20)	primary key	
beginning_date_counter	int(11)		
data_counter	varchar(50)		
variation_counter	int		
total_counter	int		

admin_table

This table is a internal cash where list admin roles are stored. It is just a cash and and it does not need to saved. You may remove its content if needed. It will just make next Sympa start slower..

Fields	Type	Properties	Usage
user_admin	varchar(100)	primary key	List admin email
list_admin	varchar(50)	primary key	Listname
robot_admin	varchar(80)	primary key	List domain
role_admin	enum('listmaster','owner','editor')	primary key	A role of this user for this list (editor, owner or listmaster which a kind of list owner too)
profile_admin	enum('privileged','normal')		privilege level for this owner, value <i>normal</i> or <i>privileged</i> . The related privilege are listed in editlist.conf.

Fields	Type	Properties	Usage
date_admin	datetime		date this user become a list admin
update_admin	datetime		last update timestamp
reception_admin	varchar(20)		email reception option for list management messages
visibility_admin	varchar(20)		admin user email can be hidden in the list web page description
comment_admin	varchar(150)		
subscribed_admin	int(1)		Set to 1 if user is list admin by definition in list config file
included_admin	int(1)		Set to 1 if user is admin by an external data source
include_sources_admin	varchar(50)		name of external datasource
info_admin	varchar(150)		private information usually dedicated to listmasters who needs some additional information about list owners

netidmap_table



Fields	Type	Properties	Usage
netid_netidmap	varchar(100)	primary key	
email_netidmap	varchar(100)		
robot_netidmap	varchar(80)	primary key	
serviceid_netidmap	varchar(100)	primary key	

conf_tables



Fields	Type	Properties	Usage
value_conf	varchar(300)		the value of parameter <i>label_conf</i> of robot <i>robot_conf</i> .
robot_conf	varchar(80)	primary key	
label_conf	varchar(80)	primary key	

subscriber_table

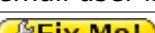
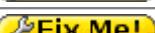
This table store subscription, subscription option etc..

Fields	Type	Properties	Usage
user_subscriber	varchar(100)	primary key	email of subscriber
list_subscriber	varchar(50)	primary key	list name of a subscription
robot_subscriber	varchar(80)	primary key	robot (domain) of the list

Fields	Type	Properties	Usage
reception_subscriber	varchar(20)		reception format option of subscriber (digest, summary, etc)
suspend_subscriber	int(1)		boolean set to 1 if subscription is suspended
suspend_start_date_subscriber	int(11)		The date (epoch) when message reception is suspended
suspend_end_date_subscriber	int(11)		The date (epoch) when message reception should be restored
bounce_subscriber	varchar(35)		FIXE
bounce_score_subscriber	smallint(6)		
custom_attribute_subscriber	text		
bounce_address_subscriber	varchar(100)		
date_subscriber	datetime		date of subscription
update_subscriber	datetime		the latest date where subscription is confirmed by subscriber
comment_subscriber	varchar(150)		Free form name
number_messages_subscriber	int(5)		the number of message the subscriber sent
visibility_subscriber	varchar(20)		
topics_subscriber	varchar(200)		topic subscription specification
subscribed_subscriber	int(1)		boolean set to 1 if subscriber comes from ADD or SUB
included_subscriber	int(1)		boolean, set to 1 is subscriber comes from an external datasource. Note that included_subscriber and subscribed_subscriber can both value 1
include_sources_subscriber	varchar(50)		comma seperated list of datasource that contain this subscriber

user_table

The user_table is mainly used to manage login from web interface. A subscriber may not appear in the user_table if he never log through the web interface..

Fields	Type	Properties	Usage
email_user	varchar(100)	primary key	email user is the key
attributes_user	text		
data_user	text		
cookie_delay_user	int(11)		
lang_user	varchar(10)		user langage preference
password_user	varchar(40)		password are stored as fringer print
gecos_user	varchar(150)		
last_login_date_user	int(11)		date epoch from last login, printed in login result for security purpose
last_login_host_user	varchar(60)		host of last login, printed in login result for security purpose
wrong_login_count_user	int(11)		login attempt count, used to prevent brut force attack

spool_table

This table is created in version 6.3. It replace most of spools on file system for clustering purpose.

Fields	Type	Properties	Usage		
messagekey_spool	bigint(20)	primary key	autoincrement key		
spoolname_spool	enum('msg','auth','mod','digest','archive','bounce','subscribe','topic','bulk','validated','task')		the spool name		
list_spool	varchar(50)				
robot_spool	varchar(80)				
priority_spool	varchar(2)		priority (list priority, owner priority etc)		
date_spool	int(11)		the date a message is copied in spool table		
message_spool	longtext		message as string b64 encoded		
messagelock_spool	varchar(90)		a unique string for each process : \$\$@hostname		
lockdate_spool	int(11)		the date a lock is set. Used in order detect old locks		
message_status_spool	enum('ok','bad')		if problem when processed entries have bad status		
message_diag_spool	text		the reason why a message is moved to bad		
type_spool	varchar(15)	list	list-request	sympa robot	other rcp
authkey_spool	varchar(33)		authentication key for email challenge		
headerdate_spool	varchar(80)		the message header date		
create_list_if_needed_spool	int(1)		set to 1 if message is related to a dynamic list, set to 0 if list as been created or if list is static		
subject_spool	varchar(110)		subject of the message stored to list spool content faster		
sender_spool	varchar(110)		this info is stored to browse spool content faster		
messageid_spool	varchar(300)		stored to list spool content faster		
spam_status_spool	varchar(12)		spamstatus scenario result		
size_spool	int(11)		info stored in order to browse spool content faster		
task_date_spool	int(11)		date for a task		
task_label_spool	varchar(20)		label for a task		
task_model_spool	varchar(40)		model of related task		
task_object_spool	varchar(50)		object of related task		

Fields	Type	Properties	Usage		
dkim_privatekey_spool	varchar(1000)		DKIM parameter stored for bulk daemon because bulk ignore list parameters, private key to sign message		
dkim_selector_spool	varchar(50)		DKIM parameter stored for bulk daemon because bulk ignore list parameters, DKIM selector to sign message		
dkim_d_spool	varchar(50)		DKIM parameter stored for bulk daemon because bulk ignore list parameters, the d DKIM parameter		
dkim_i_spool	varchar(100)		DKIM parameter stored for bulk daemon because bulk ignore list parameters, DKIM i signature parameter		
dkim_header_list_spool	varchar(500)		DKIM parameter stored for bulk daemon because bulk ignore list parameters, the list of headers included in signature		

bulkmailer_table

storage of recipients with a ref to a message in spool_table. So a very simple process can distribute them.

Fields	Type	Properties	Usage
messagekey_bulkmailer	varchar(80)	primary key	A pointer to a message in spool_table. It must be a value of a line in table spool_table with same value as messagekey_spool
packetid_bulkmailer	varchar(33)	primary key	An id for the packet
messageid_bulkmailer	varchar(200)		The message Id
recipients_bulkmailer	text		the comma separated list of recipient email for this message
returnpath_bulkmailer	varchar(100)		the return path value that must be set when sending the message
robot_bulkmailer	varchar(80)		
listname_bulkmailer	varchar(50)		
verp_bulkmailer	int(1)		A boolean to specify if VERP is required, in this cas return_path will be formated using verp form

Fields	Type	Properties	Usage
tracking_bulkmailer	enum('mdn','dsn')		Is DSN or MDM requiered when sending this message?
merge_bulkmailer	int(1)		Boolean, if true, the message is to be parsed as a TT2 template foreach recipient
priority_message_bulkmailer	smallint(10)		 Fix Me!
priority_packet_bulkmailer	smallint(10)		 Fix Me!
reception_date_bulkmailer	int(11)		The date where the message was received
delivery_date_bulkmailer	int(11)		The date the message was sent
lock_bulkmailer	varchar(30)		A lock. It is set as process-number @ hostname so multiple bulkmailer can handle this spool

exclusion_table

exclusion table is used in order to manage unsubscription for subscriber inclued from an external data source.

Fields	Type	Properties	Usage
user_exclusion	varchar(100)	primary key	
date_exclusion	int(11)		
list_exclusion	varchar(50)	primary key	

session_table

managment of http session.

Fields	Type	Properties	Usage
id_session	varchar(30)	primary key	the identifier of the database record
start_date_session	int(11)		the date when the session was created
date_session	int(11)		date epoch of the last use of this session. It is used in order to expire old sessions
remote_addr_session	varchar(60)		The IP address of the computer from which the session was created
robot_session	varchar(80)		The virtual host in which the session was created
email_session	varchar(100)		the email associated to this session
hit_session	int(11)		the number of hit performed during this session. Used to detect crawlers
data_session	text		parameters attached to this session that don't have a dedicated column in the database

one_time_ticket_table

One time ticket are random value use for authentication chalenge. A ticket is associated with a context which look like a session.

Fields	Type	Properties	Usage
data_one_time_ticket	varchar(200)		
status_one_time_ticket	varchar(60)		
remote_addr_one_time_ticket	varchar(60)		
ticket_one_time_ticket	varchar(30)	primary key	
date_one_time_ticket	int(11)		
robot_one_time_ticket	varchar(80)		
email_one_time_ticket	varchar(100)		

notification_table

used for message tracking feature. If the list is configured for tracking, outgoing messages include a delivery status notification request and optionnaly a return receipt request. When DSN MDN are received by Sympa, they are stored in this table in relation with the related list and message_id.

Fields	Type	Properties	Usage
date_notification	int(11)		 Fix Me!
pk_notification	bigint(20)	primary key	Autoincrement key
message_id_notification	varchar(100)		initial message-id. This field is used to search DSN and MDN related to a particular message
recipient_notification	varchar(100)		email address of recipient for which a DSN or MDM was received
reception_option_notification	varchar(20)		The subscription option of the subscriber when the related message was sent to the list. Useful because some recipient may have option such as <i>digest</i> or <i>nomail</i>
status_notification	varchar(100)		Value of notification
arrival_date_notification	varchar(80)		reception date of latest DSN or MDM
type_notification	enum('DSN', 'MDN')		Type of the notification (DSN or MDM)
message_notification	longtext		The DSN or the MDM itself
list_notification	varchar(50)		The listname the message was issued for
robot_notification	varchar(80)		The robot the message is related to

logs_table

Each important event is stored in this table. List owners and listmaster can search entries in this table using web interface..

Fields	Type	Properties	Usage
id_logs	bigint(20)	primary key	Unique log's identifier
user_email_logs	varchar(100)		e-mail address of the message sender or email of identified web interface user (or soap user)
date_logs	int(11)		date when the action was executed
robot_logs	varchar(80)		name of the robot in which context the action was executed

Fields	Type	Properties	Usage
list_logs	varchar(50)		name of the mailing-list in which context the action was executed
action_logs	varchar(50)		name of the Sympa subroutine which initiated the log
parameters_logs	varchar(100)		List of commas-separated parameters. The amount and type of parameters can differ from an action to another
target_email_logs	varchar(100)		e-mail address (if any) targeted by the message
msg_id_logs	varchar(255)		identifier of the message which triggered the action
status_logs	varchar(10)		exit status of the action. If it was an error, it is likely that the error_type_logs field will contain a description of this error
error_type_logs	varchar(150)		name of the error string – if any – issued by the subroutine
client_logs	varchar(100)		IP address of the client machine from which the message was sent
daemon_logs	varchar(10)		name of the Sympa daemon which ran the action

stat_table

Statistic item are store in this table, Sum average etc are stored in Stat_counter_table.

Fields	Type	Properties	Usage
daemon_stat	varchar(10)		
parameter_stat	varchar(50)		
list_stat	varchar(150)		
id_stat	bigint(20)	primary key	
operation_stat	varchar(50)		
email_stat	varchar(100)		
user_ip_stat	varchar(100)		
read_stat	tinyint(1)		
date_stat	int(11)		
robot_stat	varchar(80)		

stat_counter_table

Use in conjunction with stat_table for users statistics.

Fields	Type	Properties	Usage
list_counter	varchar(150)		
end_date_counter	int(11)		
robot_counter	varchar(80)		
id_counter	bigint(20)	primary key	
beginning_date_counter	int(11)		
data_counter	varchar(50)		
variation_counter	int		
total_counter	int		

admin_table

This table is a internal cash where list admin roles are stored. It is just a cash and and it does not need to saved. You may remove its content if needed. It will just make next Sympa start slower..

Fields	Type	Properties	Usage
user_admin	varchar(100)	primary key	List admin email
list_admin	varchar(50)	primary key	Listname
robot_admin	varchar(80)	primary key	List domain
role_admin	enum('listmaster','owner','editor')	primary key	A role of this user for this list (editor, owner or listmaster which a kind of list owner too)
profile_admin	enum('privileged','normal')		privilege level for this owner, value <i>normal</i> or <i>privileged</i> . The related privilege are listed in editlist.conf.
date_admin	datetime		date this user become a list admin
update_admin	datetime		last update timestamp
reception_admin	varchar(20)		email reception option for list managment messages
visibility_admin	varchar(20)		admin user email can be hidden in the list web page description
comment_admin	varchar(150)		
subscribed_admin	int(1)		Set to 1 if user is list admin by definition in list config file
included_admin	int(1)		Set to 1 if user is admin by an external data source
include_sources_admin	varchar(50)		name of external datasource
info_admin	varchar(150)		private information usually dedicated to listmasters who needs some additional information about list owners

netidmap_table



Fields	Type	Properties	Usage
netid_netidmap	varchar(100)	primary key	
email_netidmap	varchar(100)		
robot_netidmap	varchar(80)	primary key	
serviceid_netidmap	varchar(100)	primary key	

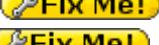
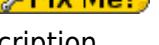
conf_tables



Fields	Type	Properties	Usage
value_conf	varchar(300)		the value of parameter <i>label_conf</i> of robot <i>robot_conf</i> .
robot_conf	varchar(80)	primary key	
label_conf	varchar(80)	primary key	

subscriber_table

This table store subscription, subscription option etc.

- user_subscriber : email of subscriber Primary key.
- list_subscriber : list name of a subscription Primary key.
- robot_subscriber : robot (domain) of the list Primary key.
- reception_subscriber : reception format option of subscriber (digest, summary, etc)
- suspend_subscriber : boolean set to 1 if subscription is suspended
- suspend_start_date_subscriber : The date (epoch) when message reception is suspended
- suspend_end_date_subscriber : The date (epoch) when message reception should be restored
- bounce_subscriber : FIXE
- bounce_score_subscriber : 
- custom_attribute_subscriber : 
- bounce_address_subscriber : 
- date_subscriber : date of subscription
- update_subscriber : the latest date where subscription is confirmed by subscriber
- comment_subscriber : Free form name
- number_messages_subscriber : the number of message the subscriber sent
- visibility_subscriber : 
- topics_subscriber : topic subscription specification
- subscribed_subscriber : boolean set to 1 if subscriber comes from ADD or SUB
- included_subscriber : boolean, set to 1 is subscriber comes from an external datasource. Note that included_subscriber and subscribed_subscriber can both value 1
- include_sources_subscriber : comma seperated list of datasource that contain this subscriber

user_table

The user_table is mainly used to manage login from web interface. A subscriber may not appear in the user_table if he never log through the web interface.

- email_user : email user is the key Primary key.
- attributes_user : 
- data_user : 
- cookie_delay_user : 
- lang_user : user langage preference
- password_user : password are stored as finger print
- gecos_user :
- last_login_date_user : date epoch from last login, printed in login result for security purpose
- last_login_host_user : host of last login, printed in login result for security purpose
- wrong_login_count_user : login attempt count, used to prevent brut force attack

spool_table

This table is created in version 6.3. It replace most of spools on file system for clustering purpose

- messagekey_spool : autoincrement key Primary key.
- spoolname_spool : the spool name
- list_spool :
- robot_spool :
- priority_spool : priority (list priority, owner priority etc)
- date_spool : the date a message is copied in spool table
- message_spool : message as string b64 encoded
- messagelock_spool : a unique string for each process : \$\$@hostname
- lockdate_spool : the date a lock is set. Used in order detect old locks
- message_status_spool : if problem when processed entries have bad status
- message_diag_spool : the reason why a message is moved to bad
- type_spool : list | list-request | sympa robot | other rcp
- authkey_spool : authentication key for email challenge
- headerdate_spool : the message header date
- create_list_if_needed_spool : set to 1 if message is related to a dynamic list, set to 0 if list has been created or if list is static
- subject_spool : subject of the message stored to list spool content faster
- sender_spool : this info is stored to browse spool content faster
- messageid_spool : stored to list spool content faster
- spam_status_spool : spamstatus scenario result
- size_spool : info stored in order to browse spool content faster
- task_date_spool : date for a task
- task_label_spool : label for a task
- task_model_spool : model of related task
- task_object_spool : object of related task
- dkim_privatekey_spool : DKIM parameter stored for bulk daemon because bulk ignore list parameters, private key to sign message
- dkim_selector_spool : DKIM parameter stored for bulk daemon because bulk ignore list parameters, DKIM selector to sign message
- dkim_d_spool : DKIM parameter stored for bulk daemon because bulk ignore list parameters, the d DKIM parameter
- dkim_i_spool : DKIM parameter stored for bulk daemon because bulk ignore list parameters, DKIM i signature parameter
- dkim_header_list_spool : DKIM parameter stored for bulk daemon because bulk ignore list parameters, the list of headers included in signature

bulkmailer_table

storage of recipients with a ref to a message in spool_table. So a very simple process can distribute them

- messagekey_bulkmailer : A pointer to a message in spool_table. It must be a value of a line in table spool_table with same value as messagekey_spool Primary key.
- packetid_bulkmailer : An id for the packet Primary key.
- messageid_bulkmailer : The message Id

- recipients_bulkmailer : the comma separated list of recipient email for this message
- returnpath_bulkmailer : the return path value that must be set when sending the message
- robot_bulkmailer :
- listname_bulkmailer :
- verp_bulkmailer : A boolean to specify if VERP is required, in this case return_path will be formatted using verp form
- tracking_bulkmailer : Is DSN or MDM required when sending this message?
- merge_bulkmailer : Boolean, if true, the message is to be parsed as a TT2 template foreach recipient
- priority_message_bulkmailer : 
- priority_packet_bulkmailer : 
- reception_date_bulkmailer : The date where the message was received
- delivery_date_bulkmailer : The date the message was sent
- lock_bulkmailer : A lock. It is set as process-number @ hostname so multiple bulkmailer can handle this spool

exclusion_table

exclusion table is used in order to manage unsubscription for subscriber included from an external data source

- user_exclusion : Primary key.
- date_exclusion :
- list_exclusion : Primary key.

session_table

management of http session

- id_session : the identifier of the database record Primary key.
- start_date_session : the date when the session was created
- date_session : date epoch of the last use of this session. It is used in order to expire old sessions
- remote_addr_session : The IP address of the computer from which the session was created
- robot_session : The virtual host in which the session was created
- email_session : the email associated to this session
- hit_session : the number of hits performed during this session. Used to detect crawlers
- data_session : parameters attached to this session that don't have a dedicated column in the database

one_time_ticket_table

One time ticket are random values used for authentication challenge. A ticket is associated with a context which looks like a session

- data_one_time_ticket :
- status_one_time_ticket :
- remote_addr_one_time_ticket :
- ticket_one_time_ticket : Primary key.

- date_one_time_ticket :
- robot_one_time_ticket :
- email_one_time_ticket :

notification_table

used for message tracking feature. If the list is configured for tracking, outgoing messages include a delivery status notification request and optionnaly a return receipt request. When DSN MDN are received by Sympa, they are stored in this table in relation with the related list and message_id

- date_notification : 
- pk_notification : Autoincrement key Primary key.
- message_id_notification : initial message-id. This field is used to search DSN and MDN related to a particular message
- recipient_notification : email address of recipient for which a DSN or MDM was received
- reception_option_notification : The subscription option of the subscriber when the related message was sent to the list. Useful because some recipient may have option such as *digest* or *nomail*
- status_notification : Value of notification
- arrival_date_notification : reception date of latest DSN or MDM
- type_notification : Type of the notification (DSN or MDM)
- message_notification : The DSN or the MDN itself
- list_notification : The listname the message was issued for
- robot_notification : The robot the message is related to

logs_table

Each important event is stored in this table. List owners and listmaster can search entries in this table using web interface.

- id_logs : Unique log's identifier Primary key.
- user_email_logs : e-mail address of the message sender or email of identified web interface user (or soap user)
- date_logs : date when the action was executed
- robot_logs : name of the robot in which context the action was executed
- list_logs : name of the mailing-list in which context the action was executed
- action_logs : name of the Sympa subroutine which initiated the log
- parameters_logs : List of commas-separated parameters. The amount and type of parameters can differ from an action to another
- target_email_logs : e-mail address (if any) targeted by the message
- msg_id_logs : identifier of the message which triggered the action
- status_logs : exit status of the action. If it was an error, it is likely that the error_type_logs field will contain a description of this error
- error_type_logs : name of the error string - if any - issued by the subroutine
- client_logs : IP address of the client machine from which the message was sent
- daemon_logs : name of the Sympa daemon which ran the action

stat_table

Statistic item are store in this table, Sum average etc are stored in Stat_counter_table

- daemon_stat :
- parameter_stat :
- list_stat :
- id_stat : Primary key.
- operation_stat :
- email_stat :
- user_ip_stat :
- read_stat :
- date_stat :
- robot_stat :

stat_counter_table

Use in conjunction with stat_table for users statistics

- list_counter :
- end_date_counter :
- robot_counter :
- id_counter : Primary key.
- beginning_date_counter :
- data_counter :
- variation_counter :
- total_counter :

admin_table

This table is a internal cash where list admin roles are stored. It is just a cash and and it does not need to saved. You may remove its content if needed. It will just make next Sympa start slower.

- user_admin : List admin email Primary key.
- list_admin : Listname Primary key.
- robot_admin : List domain Primary key.
- role_admin : A role of this user for this list (editor, owner or listmaster which a kind of list owner too) Primary key.
- profile_admin : privilege level for this owner, value *normal* or *privileged*. The related privilege are listed in editlist.conf.
- date_admin : date this user become a list admin
- update_admin : last update timestamp
- reception_admin : email reception option for list managment messages
- visibility_admin : admin user email can be hidden in the list web page description
- comment_admin :
- subscribed_admin : Set to 1 if user is list admin by definition in list config file
- included_admin : Set to 1 if user is admin by an external data source
- include_sources_admin : name of external datasource

- info_admin : private information usually dedicated to listmasters who needs some additional information about list owners

netidmap_table



- netid_netidmap : Primary key.
- email_netidmap :
- robot_netidmap : Primary key.
- serviceid_netidmap : Primary key.

conf_tables



- value_conf : the value of parameter *label_conf* of robot *robot_conf*.
- robot_conf : Primary key.
- label_conf : Primary key.

From:
<https://www.sympa.org/> - Sympa mailing list server



Permanent link:
<https://www.sympa.org/internals/index?rev=1313501810>

Last update: **2011/08/16 15:36**