

# BetterDocumentation SqlReadmeAwl

## sql/README.awl

The following is the text of sql/README.awl, based off of revision 161211.

Please feel free to edit it as much as you like to make it more useful. Periodically the version in Subversion will be updated to incorporate some of the changes.

To see the latest version in Subversion, [click here](#)

Feel free to write comments about your changes in the [Comments](#) section (at the bottom).

Using SpamAssassin Auto-Whitelists With An SQL Database

-----

SpamAssassin can now load users' auto-whitelists from a SQL database. The most common use for a system like this would be for users to be able to have per user auto-whitelists on systems where users may not have a home directory to store the whitelist DB files.

In order to activate the SQL based auto-whitelist you have to configure spamassassin and spamd to use a different whitelist factory. This is done with the auto\_whitelist\_factory config variable, like so:

```
auto_whitelist_factory Mail::SpamAssassin::SQLBasedAddrList
```

SpamAssassin will check the global configuration file (ie. any file matching /etc/mail/spamassassin/\*.cf) for the following settings:

```
user_awl_dsn          DBI:driver:database:hostname[:port]
user_awl_sql_username dbusername
user_awl_sql_password dbpassword
```

The first option, user\_awl\_dsn, describes the data source name that will be used to create the connection to your SQL server. It MUST be in the format as listed above. <driver> should be the DBD driver that you have installed to access your database (initially tested with MySQL, PostgreSQL and SQLite). <database> must be the name of the database that you created to store the auto-whitelist table. <hostname> is the name of the host that contains the SQL database server. <port> is the optional port number where your database server is listening.

```
user_awl_dsn          DBI:mysql:spamassassin:localhost
```

Would tell SpamAssassin to connect to the database named spamassassin using MySQL on the local server, and since <port> is omitted, the driver will use the default port number. The other two required options tells SpamAssassin to use the defined username and password to establish the connection.

If the user\_awl\_dsn option does not exist, SpamAssassin will not attempt to use SQL for the auto-whitelist.

One additional configuration option exists that allows you to set the table name for the auto-whitelist table.

```
user_awl_sql_table     awl
```

For an example of connecting to a PostgreSQL database, see the README file.

Requirements

-----

In order for SpamAssassin to work with your SQL database, you must have the perl DBI module installed, AS WELL AS the DBD driver/module for your specific database. For example, if using MySQL as your RDBMS, you must have the Mysql-Mysql module installed. Check CPAN for the latest versions of DBI and your database driver/module.

We are currently using:

```
DBI-1.20
Mysql-Mysql-modules-1.2219
perl v5.6.1
```

But older versions should work fine.

#### Database Schema

-----

The database must contain a table named by 'userawl\_sql\_table' (default setting: "awl") with at least these fields:

username varchar(100)	# this is the username whose e-mail is being filtered
email varchar(200)	# this is the address key
ip varchar(10)	# this is the ip key
count int(11)	# this is the message counter
totscore float	# this is the total calculated score

You can add as many other fields you wish as long as the above fields are contained in the table.

Included is a default table that can be safely used in your own setup. To use the default table, you must first create a database, and a username/password that can access that database. (See "Creating A Database", in "sql/README", if you don't have a suitable database ready.)

To install the table, use the following command:

```
mysql -h <hostname> -u <adminusername> -p <databasename> < awl_mysql.sql
Enter password: <adminpassword>
```

This will create the following table:

```
CREATE TABLE awl (
  username varchar(100) NOT NULL default '',
  email varchar(200) NOT NULL default '',
  ip varchar(10) NOT NULL default '',
  count int(11) default '0',
  totscore float default '0',
  PRIMARY KEY (username,email,ip)
) TYPE=MyISAM;
```

For PostgreSQL, use the following:

```
psql -U <username> -f awl_pg.sql <databasename>
```

Once you have created the database and added the table, just add the required lines to your global configuration file (local.cf). Note that you must specify the proper whitelist factory in the config file in order for this to work and the current username must be passed to spamd.

#### Testing SpamAssassin/SQL

-----

To test your SQL setup, and debug any possible problems, you should start spamd with the -D option, which will keep spamd in the foreground, and will output debug message to the terminal. You should then test spamd with a message by calling spamc. You can use the sample-spam.txt file with the following command:

```
cat sample-spam.txt | spamc
```

Watch the debug output from spamd and look for the following debug line:

```
[<pid>] dbg: auto-whitelist: sql-based connected to <your dsn>
```

If you do not see the above text, then the SQL query was not successful, and

you should consult any error messages reported.

This code has been tested using MySQL as the RDBMS, with basic tests against PostgreSQL and SQLite. It has been written with the utmost simplicity using DBI, and any database driver that conforms to the DBI interface and allows you to refer to a column on the right hand side of an expression (ie update foo set bar = bar + 1) should work with little or no problems. If you find a driver that has issues, please report them to the SADev list.

\*\*\*\*\*

NB: This should be considered BETA, and the interface, schema, or overall operation of SQL support may change at any time with future releases of SA.  
\*\*\*\*\*

## Comments

Please enter comments here. You can type @'SIG@ to insert your signature. – [DuncanFindlay](#) <<DateTime(2005-08-22T02:50:06Z)>>

---

As with file-based AWL, the SQL version grows without bounds. I've forgotten where I found the original recommendation, but adding another column like this:

```
ALTER TABLE awl add column {{lastupdate timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP;}}
```

and adding a cron job to periodically trim entries that haven't been touched in a while:

```
mysql spamassassin -e 'DELETE FROM awl WHERE lastupdate <= DATE_SUB(SYSDATE(), INTERVAL 2 WEEK);'
```

helps keep it under control. This is especially important if you've made efforts to put the MySQL data store on a RAMdisk of some kind...

– Kris Deugau 2009-08-27

I've used a similar method based on count as well for mysql-based AWL storage! But this is definitely a good idea for AWL.

Suggest that you also add an index to the column like this for mysql:

```
ALTER TABLE awl ADD INDEX lastupdate(lastupdate);
```

And I also have an index on count and would recommend running 4 tiers of deletions via cron:

```
DELETE FROM awl WHERE lastupdate <= (now() - INTERVAL 15 day) and count < 5;
DELETE FROM awl WHERE lastupdate <= (now() - INTERVAL 30 day) and count < 10;
DELETE FROM awl WHERE lastupdate <= (now() - INTERVAL 60 day) and count < 20;
DELETE FROM awl WHERE lastupdate <= (now() - INTERVAL 120 day);
```

– Kevin A. [McGrail](#) 2009-08-27