

# BetterDocumentation SqlReadme

## sql/README

The following is the text of sql/README, based off of revision 170078.

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).

```
Loading SpamAssassin User Preferences From An SQL Database
```

```
-----

SpamAssassin can now load users' score files from an SQL database.  The concept
here is to have a web application (PHP/perl/ASP/etc.) that will allow users to
be able to update their local preferences on how SpamAssassin will filter their
e-mail.  The most common use for a system like this would be for users to be
able to update the white list of addresses (whitelist_from) without the need
for them to update their $HOME/.spamassassin/user_prefs file.  It is also quite
common for users listed in /etc/passwd to not have a home directory, therefore,
the only way to have their own local settings would be through an RDBMS system.
```

```
Note that this will NOT look for test rules, only local scores,
whitelist_from(s), and required_score.
```

```
In addition, any config options marked as Admin Only will NOT be parsed from
SQL preferences.
```

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

```

user_scores_dsn          DBI:driver:connection
user_scores_sql_username dbusername
user_scores_sql_password dbpassword
```

```
The first option, user_scores_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.  <connection> can differ depending on which
database you are using.
```

```
For MySQL, connection should take the format
```

```
database:hostname[:port]
```

```
<database> must be the name of the database that you created to store the user
preference 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_scores_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.
```

```
For PostgreSQL, connection should take the following format:
```

```
dbname=database;[host=hostname;[port=port;]
```

```

user_scores_dsn          DBI:Pg:dbname=spamassassin;host=localhost
```

```
would do the same as the previous example.
```

```
For additional information, please refer to the DBD::* documentation
for your particular driver.
```

The spamd server will not pay attention to SQL preferences by default, even with `user_scores_dsn` set in the config files. You must startup spamd with the proper options (ie `-q` or `-Q`, see perldoc spamd for more information). If the `user_scores_dsn` option does not exist, SpamAssassin will not attempt to use SQL for retrieving users' preferences.

While scanning a message if spamd is unable to connect to the server specified in `user_scores_dsn` or an error occurs when querying the SQL server then spam checking will not be performed on that message.

The `user_scores_sql_username` and `user_scores_sql_password` options are required if your database server requires a username and password to be sent on connect.

If you have a table layout that differs from the default, please review the documentation for `user_scores_sql_custom_query` for information on how deal with a custom layout.

#### 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 and newer versions should work fine as the SQL code in SpamAssassin is as simple as could be.

#### Database Schema

-----

The database must contain a table, default name "userpref", with at least three fields:

```
username varchar(100)           # this is the username whose e-mail is being filtered
preference varchar(30)          # the preference (whitelist_from, required_score, etc.)
value varchar(100)              # the value of the named preference
```

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

Note that you can either use just the mail recipient's username for the "username" field, in which case a `varchar(8)` should suffice. Alternatively, you can use the entire recipient's email address, e.g. "user@example.com", and use the full `varchar(100)`.

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.

If you wish to use a table that differs from the included default you should review the `user_scores_sql_custom_query` config option for information on making it work correctly.

To create a database, if one does not already exist, see "Creating A Database" below.

To install the table to a mysql database, use the following command:

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

This will create the following table:

```
CREATE TABLE userpref (  
  username varchar(100) default NOT NULL,  
  preference varchar(30) default NOT NULL,  
  value varchar(100) default NOT NULL,  
  prefid int(11) NOT NULL auto_increment,  
  PRIMARY KEY (prefid),  
  INDEX (username)  
) TYPE=MyISAM;
```

For PostgreSQL, use the following command:

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

This will create a table similar to above.

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 be running spamc/spamd in order for this to work, and the current username must be passed to spamd. This can be done from spamc using the following .procmailrc recipe:

```
:0fw  
| /usr/local/bin/spamc -f
```

(watch out; spamc could be installed as /usr/bin/spamc instead.)

If you are using this from /etc/procmailrc, you must include DROPPRIVS=yes before spamc. An example /etc/procmailrc:

```
DROPPRIVS=yes  
  
:0fw  
| /usr/local/bin/spamc -f
```

Also note that spamd may need the "-q" switch so it knows to look up users in the SQL table instead of /etc/passwd. See "man spamd".

#### Creating A Database

-----

Here's the command to create a MySQL database, and user/password pair to access it:

```
mysql -h <hostname> -u <adminusername> -p  
Enter password: <adminpassword>  
mysql> use mysql;  
mysql> insert into user (Host, User, Password) values('localhost', '<username>', password('<password>'));  
mysql> insert into db (Host, Db, User, Select_priv) values('localhost', '<databasename>', '<username>', 'Y');  
mysql> create database <databasename>;  
mysql> quit
```

NOTE: If you intend to use this database for Bayes and/or AWL data you may need to grant additional privs (ie Insert\_priv, Update\_priv and Delete\_priv). Please refer to the MySQL documentation for the proper method of adding these privs.

To create the database for PostgreSQL, with a username/password:

```
psql -U <adminuser> templatel  
templatel=# CREATE USER <username> PASSWORD '<password>';  
templatel=# CREATE DATABASE <databasename> OWNER = <username>;
```

#### 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:

```
retrieving prefs for <username> from SQL server
```

If you do not see the above text, then the SQL query was not successful, and you should see any error messages reported. <username> should be the user that was passed to spamd and is usually the user executing spamc.

Note that under the default configuration any prefs stored under the username '@GLOBAL' are used as defaults for all users.

This code has only been tested using MySQL as the RDMS, but it has been written with the utmost simplicity using DBI, and any database driver that conforms to the DBI interface should work without problems.

\*\*\*\*\*

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.

\*\*\*\*\*

#### Web Interfaces

-----

Several web interfaces have been created for per user configurations. You can find more information about these on the SpamAssassin wiki: <http://wiki.apache.org/spamassassin/WebUserInterfaces>

#### Conversion Utility

-----

If you are migrating from a setup where your users already have preferences set up in preferences files, you may find Dan Mahoney's updateprefs.pl script helpful. It can be found here: <http://www.gushi.org/updateprefs.pl>

## Comments

Please enter comments here. You can type @'SIG@ to insert your signature. – [DuncanFindlay](#) <<DateTime(2005-08-22T02:47:07Z)>>

Dan's script was originally mentioned in May 2005 [here](#), but he hasn't had much reaction to it – perhaps very few people have migrated to SQL user\_prefs? Maybe his script should be put in the distribution under contrib/? – [TimHunter](#) <<DateTime(2006-12-12T17:53:29Z)>>