

# Login Intergration with 3rd Party System

Contributed by Administrator  
Monday, 18 December 2006

AST has the ability to read usernames and passwords from 3rd party SQL databases. For example, AST can use the login tables for Joomla or Pligg, allowing your users to log in once and get access to AST. This article describes how to integrate AST with your existing system.

## Setting up AST

The first thing you will need is an AST executable that has been built with SQL support. To do this, pass the configure flag `--enable-sql` during the configure step when compiling AST. Also, due to bugs in MyODBC 3.5 you will need to pass `--enable-mysql` if you are using MySQL.

If you are using Pligg or Joomla, these are currently supported natively by AST. If not, you will need to determine the table and fields used by your software. You also need to figure out how it encodes its passwords. You can usually do this by looking at the source code for the login page. Most software uses MD5 or SHA-1, possibly with some salt.

To setup SQL-based authentication, go to Setup -> System Settings -> Authentication in AST. Enter the information about your database and save it. Log out of AST and try logging in using an account from your database. Note: You MUST have a user called admin your database, otherwise you won't be able to access the administrator account of AST.

If you can't log in to AST anymore then look in the file `var/log/sqlauth_log` for information on why it failed. If you need to make changes to your settings then run `./ast --reset-defauth-auth` Then you can log in using the administrator account built into AST and change your settings.

## Integration with your Site

At this point you can log into AST from your existing site. However, you may also wish to log your users into AST when they log into your site. To do this you need to set a cookie for AST in your login page. You can figure out what the cookie should be by running `ast --session-key username`. The cookies you should set are `ast-username` and `ast-session-key` where `ast-session-key` is the key you obtained by running `ast`. Here is some PHP code to do this:

```
$astpipe = popen("ast/ast --session-key $username", "r");
$sesskey = fread($astpipe,2000);
$sesskey = rtrim($sesskey);
if( pclose($astpipe) != 0 ) { /* error */ }
setcookie('ast-session-key', $sesskey, null, '/ast/ ');
setcookie('ast-username', $username, null, '/ast/ ');
```

When setting the the path for the cookie make sure it is where `ast` is located and that it contains the trailing `/`. One other potential problem is that on some systems PHP is run under a different user. To fix this problem you can make AST `setuid` (`chmod 6755 ast`). If your web server doesn't run `setuid` programs then copy `ast` to a new file (say `ast-php`) and make that file `setuid` (continue to use `ast` for web applications).