

Description

This module allows users to log in through external database validation. If a user is not known, an account will be created. Optionally synchronize the Security Roles for the user. (MS SQL and MySQL).



Module settings

Connection string: Input the connection string to connect to the external database.

A connection string should look something like this: *Data Source=192.168.0.55; Initial Catalog=DotNetNuke; User ID=DnnUser; Password=1A2b3C4d*

SQL query: The query to check the username and password in the external database. This could also be a stored procedure

External database is MySQL: Check if the external database is MySQL.

Test the connection: Fill out a username and password to test the connection and query. Then click the “test login” link to see if the login works.

Label position: Select the placement of the labels for the textboxes (above or left). Default is above.

Redirect after login: Select the page in the portal where the user should be redirected to after a successful login. If there is no page specified, the module will try to redirect to the redirect page configured in the portal settings.

Enable the “Stay logged in” checkbox: Enable this option if you want to allow users to keep themselves logged in the next time they visit the site.

For this option to work, the following key must be changed in the Web.config file of your DNN installation if this is not already done so: `<add key="PersistentCookieTimeout" value="0" />`

You must set a value that is greater than 0. This value is in minutes, so allowing a user to stay logged in for a maximum of 30 days, you must set this value to 43200.

Header Text: This text will be displayed at the top of the module. You can use this for extra instructions or other information.

Footer Text: This text will be displayed at the bottom of the module.

Basic Settings

Please read [the manual](#) for query and stored procedure instructions and options.

Connection string: ⓘ

SQL query: ⓘ

```
SELECT 'Test' AS FIRSTNAME, 'Account' AS LASTNAME FROM
vdwwd_testlogin WHERE username = [USERNAME] AND password =
[PASSWORD]
```

External database is MySQL

Test username:

Test password:

[Test login](#)

Label position: ⓘ

Redirect after login: ⓘ

Enable the "Stay logged in" checkbox

Header text ⓘ

Footer text ⓘ

Security Role Settings

Add security role: Optionally select a existing security role to add to the user when the login is successful.

Security role SQL query: The query to check the roles for the user in the external database. This could also be a stored procedure. When a role does not exists it will be created.

Security Role Settings

Add security role: ⓘ

Security role SQL query: ⓘ

```
SELECT department AS ROLE FROM vdwwd_testroles WHERE
username = [USERNAME]
```

CSS Classes

You can define your own CSS classes in the settings for the module for the TextBox, Button and Validator controls.

TextBox Class: ⓘ

Button Class: ⓘ

Validator Class: ⓘ

Label Class: ⓘ

SQL Query options

To create a successful login query, the following tokens (case sensitive) can be used in the query: [USERNAME] and [PASSWORD]. The query or stored procedure MUST return a FIRSTNAME and LASTNAME field.

```
SELECT user_firstname AS FIRSTNAME, user_lastname AS LASTNAME FROM users WHERE (user_username = [USERNAME]) AND (user_password = [PASSWORD])
```

OR

```
Login_Stored_Procedure [USERNAME], [PASSWORD]
```

The following optional fields the query can return could be used to synchronize the user information: EMAIL, STREET, CITY, COUNTRY, POSTALCODE and TELEPHONE. Your query could look something like this:

```
SELECT user_firstname AS FIRSTNAME, user_lastname AS LASTNAME , user_email AS EMAIL, user_city AS CITY FROM users WHERE (user_username = [USERNAME]) AND (user_password = [PASSWORD])
```

Check if the query ONLY returns a result when the username and password are correct. Try to avoid something like this:

```
SELECT user_firstname AS FIRSTNAME, user_lastname AS LASTNAME FROM users WHERE (user_username = [USERNAME]) OR (user_password = [PASSWORD])
```

The above example is incorrect. It will return more than 1 result if 2 or more users have the same password.

When using a stored procedure you can return an ERROR token when something is wrong in the external database. This will be displayed in the module as an error message. This could look something like this:

```
SELECT 'Custom error message' AS ERROR
```



Security role SQL Query options

In the settings of the module you can select an existing security role that will be added to a user when he is successfully logged in. If this role is changed by an admin, the old role will be deleted and the new role will be added.

The Administrator role will not be added or removed by this module for security purposes. This role has to be added to a user using the DNN user management.

Or you can use a custom query or stored procedure to synchronize security roles. The following token (case sensitive) can be used in the query: [USERNAME]. The query or stored procedure MUST return a ROLE field.

```
SELECT department AS ROLE FROM company_departments WHERE (user_username = [USERNAME])
```

or

```
SELECT department AS ROLE FROM company_departments
```

OR

```
UserRole_Stored_Procedure [USERNAME]
```

When a role does not exist within DNN it will be created. Then the new role(s) will be added to the user. When an existing user logs in and he no longer has an existing role, it will be deleted.

Notes

Passwords are NOT kept in the website database for security. These users have to login through the EasyLogin module to authenticate. Also make sure they cannot change their password through the DNN profile page.

Make sure the query or stored procedure only returns 1 result, otherwise login will fail.

Connecting to an external database could be blocked by your hosting provider. Make sure the server and/or firewall will allow connections to external databases.

The password is sent unencrypted to the external database, so make sure the connection is secure.

NEVER store passwords unencrypted in a database!