

Using Remote Authentication

Remote authentication allows you to integrate your organization's authentication system with KBPublisher.

Before you start:

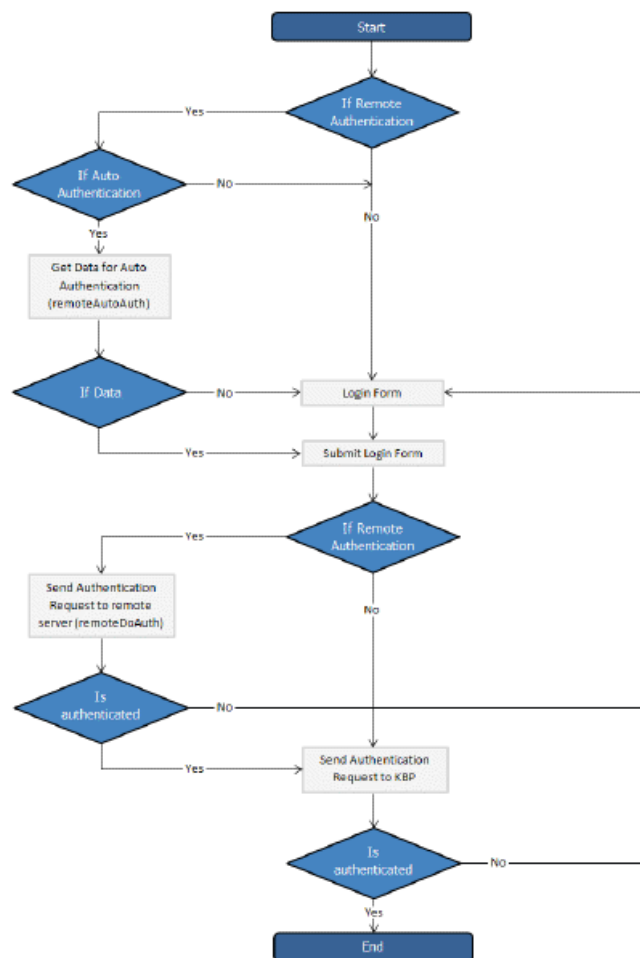
- We assume that you have some experience with PHP and with the system you are connecting to.

Steps to enable Remote Authentication

- Click on **Settings -> Authentication Provider -> Remote**
- Check **Enable Remote Authentication** checkbox (make sure that `$conf['auth_remote']` in the file `admin/config.inc.php` is set to 1)
- Set required values for constants in file `admin/lib/custom/remote_auth.php`
- Customize the `_remoteDoAuth` function in the file `admin/lib/custom/remote_auth.php` to authenticate the username and password passed to it against your own authentication system
- Rename the function `_remoteDoAuth` to `remoteDoAuth`

Quick summary of the process

- End user goes to site
- Remote Authentication checks for valid user credentials
 - If auto-authentication is set, does this automatically
 - If auto-authentication is not set, user logs in first
- KBPublisher authenticates the user .



Customizing the remoteDoAuth function

In your installation there is a folder `admin/lib/custom`. Within that folder is a file called `remote_auth.php`. This file contains the `_remoteDoAuth` function. Customize this function to do authentication against your internal system by using the username and password provided.

Here is a simple example of the function customized to authenticate against a MySQL database:

```
function remoteDoAuth($username, $password) {  
  
    $user = false;
```

```

$db = &DBUtil::connect($conf);

$sql = "SELECT
    id AS 'remote_user_id',
    email, username, first_name, last_name
FROM your_remote_users_table
WHERE username = '%s' AND password = '%s'";
$sql = sprintf($sql, $username, $password);
$result = $db->Execute($sql) or die(DBUtil::error($sql, true, $db));

// if found
if($result->RecordCount() == 1) {
    $user = $result->FetchRow();
    $user['password'] = $password; // here you should provide not md5ing password

    // assign a priv to user (optional)
    // it is fully up to you how to determine who is authenticated and what priv to assign
    // set to off to not rewrite on login
    $user['priv_id'] = 'off';

    // assign a role to user (optional)
    // it is fully up to you how to determine who is authenticated and what role to assign
    // set to off to not rewrite on login
    $user['role_id'] = 1;
}

return $user;
}

```

Also see examples in [attached files](#).

Tracking logins

You can see how your remote authentication works in logs Logs/Logins

For debugging every last login is logged to a file called *last_remote_login.log* in the KBPublisher cache directory (*APP_CACHE_DIR* in *admin/config.inc.php*).

For example: */home/username/ kb_cache/last_remote_login.log*

Article ID: 379

Last updated: 24 Aug, 2022

Updated by: Leontev E.

Revision: 7

User Manual v8.0 -> Single Sign On -> Remote Authentication -> Using Remote Authentication

<https://www.kbpublisher.com/kb/entry/379/>