

Deploying Single Sign-on

Instead of storing separate user accounts and passwords in the CommonSpot database, you can instead authenticate a user account against a windows AD domain, LDAP server, or custom database.

The key to linking your authentication scheme to an external system is to create a file named custom-authentication.cfm in the root of your web site.

Custom-authentication.cfm is invoked after CommonSpot has compared a users' ID and password against the commonspot-users.users table. It is automatically passed in the following variables:

Variable	Description
Attributes.userid	The user account entered in the login dialog
Attributes.password	The password entered in the login dialog
Attributes.csAuthenticated	Boolean – true if the user authenticated successfully against commonspot-users
Attributes.UserExists	0 if the user id was not found in commonspot-users, otherwise it is set to the numeric id of the user account
Attributes.profileExists	0 if no record exists in the commonspot-users.contacts table

In order to signal CommonSpot that the user's account is valid, you must include the following code:

```
<cfset "caller.#attributes.isauthenticatedvar#" = 1>
<cfset "caller.#attributes.defaultuseridvar#" =
    attributes.userid>
```

The first line of code signals that the user account is valid. The second line of code instructs CommonSpot that the user id to utilize for logging the user into the system is what was entered by them on the login page. Note that you could, in theory, use this feature to log a number of users in under the same generic account name.

If the value that is passed to caller.#attributes.defaultuseridvar# is not found in the commonspot-users.users table, CommonSpot will automatically create a record for you. However, for this account to be edited via the CommonSpot Administrator, you must also create code to populate a related row in the commonspot-users.contacts table.

You can also use the variable session.user.requestTarget to route the user to a specific url after they have logged-in.

Sample custom-authentication.cfm file

The following sample custom-authentication.cfm file authenticates the user against an external LDAP server. Note that you could alternatively use the <cfntauthenticate> tag instead of <cfldap> to authenticate against a windows domain in which the ColdFusion server was a member.

```

<cfset bauthenticated = 1>

<cfset server="myADserver.mydomain.com">
<cfset ADdomain = "flsdc">
<cfset Domain = "figleaf.com">
<cfset ADStub = "Fig Leaf Users">

<!--- perform NT authentication --->
<cftry>
  <cfldap action="QUERY"
    name="auth"
    start="OU=#adStub#,DC=#listgetat(domain,1,".")#,
DC=#listgetat(domain,2,".")#"
    attributes="cn,dn,displayname,sAMAccountName"
    scope="SUBTREE"
    filter="sAMAccountName=#attributes.userid#"
    server="#server#"
    username="#mydomain#\#attributes.userid#"
    password="#attributes.password#"
  >

  <cfcatch type="application">
    <cfset bauthenticated = 0>
  </cfcatch>
</cftry>

<cfset "caller.#attributes.isauthenticatedvar#" =
                                     bauthenticated>
<cfset "caller.#attributes.defaultuseridvar#" =
                                     attributes.userid>

<cfscript>
  // redirect user to their portal page
  if (not structkeyexists(session.user,"requesttarget")) {
    session.user.requesttarget = structnew();
    session.user.requesttarget.targeturl = "someurl.cfm";
  }
</cfscript>

```