LDAP Tracker Field

Display a field value from a specific user, computer or other object in LDAP. The operative word here being *display*. In order to make this work, you first have to configure a Data Service in Tiki.

Create a DSN

To set up your instance of Tiki to use this field type, first navigate to the "Admin Content Sources" page which can be found in text at the bottom of the admin page under "Crosslink to other features and settings > Text area features > DSN" (path is /tiki-admin_dsn.php from the root of your instance).

Under "Create/edit DSN" provide a name for the data service. This could be whatever you want. You will use this in the tracker to refer to the LDAP source (You could just use "LDAP" for example).
For the DSN, the format will depend upon what kind of directory you intend to query. For Microsoft Active Directory, use this format:

LDAP DSN

```
ldap://binduser:password@servername.contoso.com/ou=Users,dc=contoso,dc=com
```

where:

1. binduser: a user account *cn* with read access to your directory service (if you're using Active Directory, it generally requires authentication in order to return any response)
2. password = the password for that account (keep in mind that only Tiki administrators can see this page)
3. servername: the directory server (if you have several, pick one)
4. contoso.com: example domain - adapt for your environment
5. ou=Users,dc=contoso,dc=com: An example of the LDAP path where user objects are stored. The search scope is subtree, so keep this in mind when specifying the path. Note that if you use Containers instead of Organizational Units, this example would begin with "cn=Users".

It's probably a best practice to create a service account with a complex password specifically for the purpose of your Tiki instance binding to LDAP . This account only needs guest privileges.

LDAP tracker field configuration

Once you have established your DSN, add an LDAP tracker field to your tracker and configure the "Options for LDAP" section as follows:

1. **Filter:** The instructions underneath this setting read "LDAP filter, can contain the %field name% placeholder to be replaced with the current field's name". What this actually means is that you need another field in the same tracker that contains some information that you have about the LDAP user. It could be *cn* or *mail* or *displayName* or any other attribute that is unique to that user. Using LDAP syntax (refer to LDAP Query Basics for a guide), insert the name of this field into the search like this:

   ```
   (cn=%User ID%)
   ```

   where "%User ID%" references the contents of the tracker field named "User ID" in the same tracker as this LDAP field. Another example would be:

   ```
   (|(mail=%Email%)(userPrincipalName=%Email%))
   ```

   where you're searching either the *mail* LDAP field or the *userPrincipalName* LDAP field for information stored in a tracker field named "Email" in the same tracker. Your query could be more complex, but keeping it simple gives you faster results.
2. **Field:** The instructions underneath this setting read "Field name returned by LDAP". This is *company*

or *telephoneNumber* or whatever field in LDAP that you want this particular tracker field to show for this user.
3. **DSN:** The instructions underneath this setting read "Data source name registered in Tiki". That's the name you gave to the DSN you created earlier ("LDAP" in our example).

How it works

Once the tracker queries LDAP for the information, it will display it in Tiki in real time, which is to say that what you see in your tracker list or plugin display is the current live LDAP data, not the cached information that was polled when the record was created. If the data gets updated in LDAP, it's instantly updated in Tiki wherever you display the information.

The catch is that Tiki does not store this information in its MySQL database unless you edit the individual tracker record and save it. So if you are trying to sort tracker data based on an LDAP field, you will find that sorting doesn't work - until you update every single record of that tracker to store the LDAP data in Tiki's MySQL database. If your records have multiple LDAP fields, all the current query results get stored in MySQL when you save the record. If the LDAP data changes, Tiki will display the updated information correctly but will still sort based on the last saved version that's in its database. Re-saving the record in the tracker will fix the sorting problem.

A work-around to this problem is to export all the records from your tracker containing LDAP fields, and then import them back into the tracker. The export process performs LDAP queries and includes the current findings in the exported CSV file.

See also:

- https://gitlab.com/tikiwiki/tiki/-/blob/master/lib/core/Tracker/Field/Ldap.php?ref_type=heads
- https://tiki.org/forumthread44317-LDAP-Tracker-Field-How-does-it-work