11.3 Connecting TDI to LDAP

TDI has a rich set of LDAP facilities. We will use the LDAP client connector.

1. Add an LDAP Connector after the *WriteLDIF* connector. Call it *CorpLDAP* and put it in Update mode.

2. Configure the basic connection parameters, then click the *Contexts* button to get the list of suffixes that the server supports. Select
   `dc=example,dc=org`

3. Use the data browser on the Output Map tab to check the connection.

4. Close the data browser connection, and delete all attributes from the schema panel to avoid confusion in the next step.

5. Add the following attributes to the output map:
   `$dn carLicense cn givenName objectClass postalCode sn`
   `telephoneNumber uid`
   (If you had left the discovered attributes alone in the previous step you would have to scroll more times to do this).

6. Set the Link Criteria to match on *cn* (common name).

7. Save the config and run the AL. It fails, so read the first few lines of the error report.

8. The error report complains about an undefined operation code, which is not very helpful, but it does mention the attribute concerned: *carLicense* which came from the XML parsing connector. Go to the input map of the *LookupCarNumber* connector and scroll the schema display horizontally to see all the data. Compare this with the schema for the *ReadCSV* connector: you will see that they return objects with different Java classes. This has not mattered so far, but the LDAP connector is capable of making use of extra information in AttributeValue objects, and the XML parser has supplied such an object without the extra fields.

9. The simple solution is to convert the offending attributes to strings, so go to the input map of the *LookupCarNumber* connector. Double-click on the *Assignment* field for each attribute in turn and '`.getValue()`' to the end of each expression:

   ```
   conn.carReg.getValue()
   conn.employee.getValue()
   ```

   This causes the expressions to return the first (and only) value as a `java.lang.String` object.

10. Save the AL and run it again. A different error will stop the AL. This time it is a schema violation, complaining that *uidNumber* is not allowed in the entry. This seems odd, as we have not created such an attribute in the AL. The DN of the offending entry is given, so have a look at it in Apache Directory Studio.

11. You will find that this is the 'Caroline Dobson' entry, which was in the LDAP store before the AL was run. It has been renamed as the LDAP