Starting with DAS

Unable to render {include}
The included page could not be found.

Starting with DAS

This page provides links to DAS samples and explains a simple "how to" and gives step-by-step instructions to check one of the DAS features - ColumnConverter. Other DAS features can be tried on similar lines.

Many sample examples demonstrating different DAS features are available on https://svn.apache.org/repos/asf/incubator/tuscany/java/das/samples e.g.

1. companyweb-webapp - tomcat based web sample
2. customer - J2SE based standalone application

Check readme files e.g. For companyweb https://svn.apache.org/repos/asf/incubator/tuscany/java/das/samples/company-webapp/readme.htm
for basic steps on how to get the sample running.

Once the sample is running, parts of sample can be modified based on the requirement or new code can be added from-scratch. Many features supported by DAS are listed in User Guide - http://incubator.apache.org/tuscany/rdb-das-user-guide.html

This document details the changes needed to different part of the web sample (companyweb) to experiment with more features of DAS than just basic CRUD.Let us take example of checking how to use ColumnConverters. For details on feature testing, check Tests under https://svn.apache.org/repos/asf/incubator/tuscany/java/das/rdb/src/test/

1> As a first step you need to implement **Converter** interface in a class to achieve the required column conversion. Check org.apache.tuscany.das.rdb.test.mappings.StringObfuscationConverter from the svn repository for example of converter. Make your converter class available in (Tomcat root)webapps/companyweb-webapp/WEB-INF/classes.

2> DAS functions based on external Configuration, e.g. CompanyConfig.xml file in companyweb sample. The purpose of this Config file to supply information for DataSource connection, Commands (SQL) (that DAS can execute against Database) and Database schema like - Table/Columns, Relationship and so forth. Please check Architecture Guide for complete details about Config.xsd ()
at http://incubator.apache.org/tuscany/rdb-das-architecture-guide.html. Modify Config to get converter working. The <ConnectionInfo> element should match the resource name from server.xml for DataSource.Add a <Table> which has a column with converter class name of the class you just created. Add a <Command> with kind="Select" for this table/column e.g.

```
<Command name="getFirstCustomer" SQL="Select * from CUSTOMER where ID = 1" kind="Select"/>

<Table tableName="CUSTOMER">
<Column columnName="ID" primaryKey="true"/>
<Column columnName="LASTNAME" converterClassName="org.apache.tuscany.das.rdb.test.mappings.
StringObfuscationConverter"/>
</Table>
```

3> Now the code and config setup is complete. What remains is calling the new command from .jsp and servlet. Follow the technique similar to CompanyClient.java and Company.jsp to call the new command. The returned results will verify that the converter is called and the column value from database is converted based on converter logic.
4> For more details check the test cases from ConverterTests in https://svn.apache.org/repos/asf/incubator/tuscany/java/das/rdb/src/test/.

Assumption: Tomcat version 5.5.* or 6.0.10 , Derby version 10.1.2.1.

Troubleshooting Checkpoints: Check the following and if still having issues , report at http://incubator.apache.org/tuscany/issue-tracking.html

1. All libraries are present in (Tomcat root)/webapps/companyweb/WEB-INF/lib (these libs can be taken from companyweb sample or can be downloaded from web)
   i) common-(latest version).jar
   ii) ecore-(latest version).jar
   iii) ecore-change-(latest version).jar
   iv) ecore-xmi-(latest version).jar
   v) log4j-(latest version).jar
   vi) sdo-api-xxx.jar
   vii) tuscany-das-rdb-xxx.jar
   viii) tuscany-sdo-xxx.jar
   ix) xsd-(latest version).jar
   x) derby driver jar - or whichever database you are planning to use
2. The Database exists with required tables and data.
3. WEB-INF/web.xml has entry under welcome-file-list for required .jsp like -

```
<web-app>
    <display-name>Tuscany DAS sample Company  WEB</display-name>

    <welcome-file-list id="WelcomeFileList">
        <welcome-file>Company.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

4. META-INF/context.xml has entry for the datasource the sample is using like -

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/DAS Stand alone app" debug="5" reloadable="true" crossContext="true">
   <Manager pathname=""/>
   <ResourceLink name="jdbc/dastest" global="jdbc/dastest" type="javax.sql.DataSource"
/>
</Context>
```

5. (Tomcat root)/conf server.xml has entry similar to below for the datasource -

```
<Resource name="jdbc/dastest"
      type="javax.sql.DataSource"  auth="Container"
      description="Derby database for DAS Company sample"
      maxActive="100" maxIdle="30" maxWait="10000"
      username="" password=""
      driverClassName="org.apache.derby.jdbc.EmbeddedDriver"
      url="jdbc:derby:c:\apache-tomcat-5.5.20\Databases/dastest;create=true"/>
```