WorkingWithRelationships

As mentioned elsewhere in this guide, the RDB DAS provides a simple, implicit mapping from SDO Types/Properties to Relational Tables/Columns. So, in the absence of configuration to the otherwise, reading a table named CUSTOMER with columns NAME and ADDRESS will result in a data graph containing SDO DataObjects of Type CUSTOMER with properties NAME and ADDRESS.

This is great for reading entities from a single table but clients often need to read data from related tables for example Customers and their related Orders. The RDB DAS can work with related database tables and reads that span tables produce graphs of related data objects. Continuing with our Customer->Order example, a client can read a set of Customers and related Orders by using a SELECT statement that includes a join as in the following example:

```
   DAS das = DAS.FACTORY.createDAS(getConfig("CustomerOrderConfig.xml"), getConnection());
   // Read some customers and related orders
   Command select = das.createCommand("SELECT * FROM CUSTOMER LEFT JOIN ANORDER ON CUSTOMER.ID = ANORDER.
CUSTOMER_ID WHERE CUSTOMER.ID = 1");

   DataObject root = select.executeQuery();
   DataObject customer = root.getDataObject("CUSTOMER[1]");
   List orders = customer.getList("orders");
```

You can see that the client can work with the data graph as a set of related objects. Once the application has a handle to a customer, it can reference that customers related orders by accessing the customers "orders" property.

Notice that the example provides a config file as part of creating the DAS. This file contains mapping information that defines to the DAS how the queried tables are related. Here are the contents of the config file:

```
   <Config xsi:noNamespaceSchemaLocation="http:///org.apache.tuscany.das.rdb/config.xsd" xmlns:xsi="http://www.
w3.org/2001/XMLSchema-instance">

     <Table tableName="CUSTOMER">
          <Column columnName="ID" primaryKey="true"/>
     </Table>

     <Table tableName="ANORDER">
          <Column columnName="ID" primaryKey="true"/>
     </Table>

     <Relationship name="orders" primaryKeyTable="CUSTOMER" foreignKeyTable="ANORDER" many="true">
        <KeyPair primaryKeyColumn="ID" foreignKeyColumn="CUSTOMER_ID"/>
     </Relationship>

   </Config>
```

This config file specifies the primary keys of the related tables and then the relationship between them. It also provides a convenient name for the relationship. In short, this config specifies that the CUSTOMER table and ANORDER table are related via ANORDERs column named "CUSTOMER_ID" which references CUSTOMERS column named "ID". The name of the relationship is "orders"

As a side note, the config in this example is not required since the table definitions follow a "convention" recognized by the DAS. See WorkingWithConventions.