ExternalData.DOC: How to access an external relational database

Bevan Clarke

Introduction

This note describes how to enhance a Jade schema with additional classes which permit that schema to access and manipulate an external relational database. The example given shows accessing a Microsoft ACCESS database. It is necessary to add an appropriate OBDC driver to the operating system. A Jade Wizard will guide you through the process of defining the new External Object classes and will write the method needed.

Part (a) Setup and connection

1. Exit from JADE to the Operating system. From the **Start** button find and run the **ODBC Data Source Administrator** in the full JADE folder.
2. Choose its **System DSN** tab and add MS ACCESS driver. Click **Finish**.
3. You will then need to supply

Data source name	BONZA database	(Your own choice of text)
Description	New booksellers data	(Your own choice of text)

4. Press **Select**, browse to the C:\Jadebook\Resources folder and select NEWBOOK-SELLER.MDB. Click a series of **OK**s to close the **ODBC Data Source Administrator**.

5. Re-enter JADE and its **Schema Browser**. Make active the schema that you want to empower to use the BONZA database.
6. **Browse | External Databases**. **Databases | Add**. You are now on the first of the 12 screens of the **External database Wizard**. Proceed through the screens.

   a) The Wizard's **Name** tab: We chose to write in ExternalDBfromBONZA. **Next**.

   b) The **Data source** tab: Here we are offered alternative ways of linking to a data source. We chose BONZA database from the dropdown list. **Next**.

   c) The **Tables** tab: Here we see the list of tables that the driver can find in that database. We chose only the two base tables, Baddresses and NewBookSellers. **Next**.

   d) The **Identifiers classes** tab: We set our options as follows:

   - **Class Name Prefix** . . . Ext_
   - **Use singular table names for classes**.
   - **Array name suffix** . . . Ary.
   - **Dictionary name suffix** . . . Mkd.
   - **Include keys in name**.
   - **Prefix keys with** . . . By.
   - **Set name suffix** . . . Set.

   Click **Next.**

   e) The **Define classes** tab: We see that BAddresses and NewBookSellers have become the classes Ext_Baddress and Ext_Newbookseller. Click **Add All** then **Next**.

f) The **Define Attributes** tab: The Wizard proposes attribute names based on the relational database's field names. For each table (= class) in turn . . .

1. Uncheck any fields (= attributes) that you know you will not need [we kept them all].
2. Alter the proposed attribute names if you wish. [We left them all unchanged].
3. Click the **Save** button.

Click **Next**.

g) The **Define collections** tab: Uncheck any proposed collections that you realise your application will not use. You need not leave any checked if your application needs none.

Click **Save**. The chosen collection classes will go bold.. Click **Next**.

h) The **Define references** tab: These two classes share the name key. (So a reference based on name has been proposed.) We agree so click on the dotted line and click **Add**. **Next**.

i) The **Relationship dialog**: The relationship between a NewBookSeller and her Baddress should be a simple one-to-one relationship.[1] Click **OK**.

j) The **Class query** tab: If a simple selection of all NewBookSeller or all BAddress instances were needed this screen shows the SQL commands that would generate it. For each class . . .

1. Edit the proposed SQL queries, if necessary.
2. Click **Save**.

Click **Next**.

k) The **Collections query** tab: If a selection of all NewBookSeller or all BAddress instances all BAddress instances were needed ordered on the key field name this screen shows the SQL commands that would generate it. For each class . . .

3. Edit the proposed SQL queries, if necessary.
4. Click **Save**.

Click **Next**.

l) The **References query** tab: No action needed. **Next**.

m) The **Finished query** tab: Done! Click **Finished** and JADE will generate the supporting code to allow your schema to access this external relational database as though it were an internal JADE object-oriented database. Click **Finished**.

Part (b) Accessing and updating the external database

Task #1: In this first task your jade schema will merely access the external ACCESS relational database so that the Jade schema may iterate through the records of NEWBOOKSELLERS displaying the names with the Jade write statement.

Task #2: In this second task you must alter your schema so that a menu-click will (a) open the NEWSELLERS.MDB file and (b) incorporate the new *BookSellers* into the library collection. You may wish to join the two ACCESS files first. Consider pirating much of the method code presently in **.loadBooks0ffFile.**

[1] An **One-to-one** relationship could not possibly be an ***Array*** or a ***MemberKeyDictionary*** so the harmless option ***Set*** is pre-selected. (a set with one member!)

Proof that the link is working:

Open MS ACCESS at the same time as Jade and set it to viewing ACCESS's NEWBOOKSELLERS table. The rows correspond to objects. The columns correspond to attributes.

In Jade, if you alter an attribute of an E_NewBookSeller object you will see the same alteration made to the corresponding cell of the ACCESS table and vice versa. (You may have to change the current row in ACCESS to cause its display to refresh.)

The process is illustrated in the schema \Demos\Dm49_SQL_in_Jade.