Stored Procedures

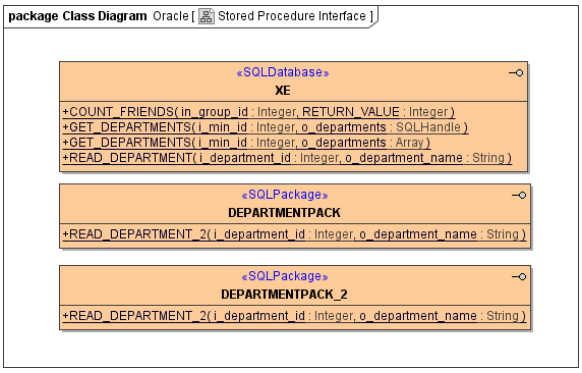
This page explains the **SQL Adapter** in Bridge context. If you were looking for the same information regarding the PAS Designer, refer to SQL Adapter in the Designer guide.

Basically, stored procedures are modeled like static UML operations that can be called using a call operation action having the stereotype <<SQLAdapter>> as depicted in the activity diagrams below.

Example File (Builder projectAdd-ons/SQL):

<your example path>\Add-ons\SQL\uml\sqlStoredProcedures.xml

The example contains the following interface that describes six stored procedures – two of them nested in the SQL packages **DEPARTMENTPACK** and **DEPARTMENTPACK_2**. If the static operations are directly owned by the <<SQLDatabase>> class, they are in the global namespace. If they are owned by a <<SQLPackage>> interface, they are owned by a SQL package having the same name as the interface.



As input and output of stored procedures, the Bridge supports all simple base types.

When specifying **String** objects as **CLOB** parameters, you have to mark them with tag **nativeType= CLOB**.

Please note that the user must be authorized to select data from the table `mysql.proc` to use stored procedures with MySQL, because the SQL API calls this table to find meta data.

Handling Stored Procedure Output Containing Multiple Records

Additionally, output values can also be of type **Array** or **SQLHandle**, see both **GET_DEPARTMENTS** procedures above. In both cases, the SQL stored procedures have the same signature on the database, they return a **cursor**. For example, in Oracle such a stored procedure may look like:

```
create or replace procedure "GET_DEPARTMENTS"
(i_min_id IN NUMERIC, o_departments OUT Types.cursor_type)
is
begin
 open o_departments FOR
    SELECT *
    FROM DEPARTMENTS
    WHERE DEPARTMENT_ID >= i_min_id;
end;
```

Depending on the **UML** output type of **o_departments**, the SQL adapter will return

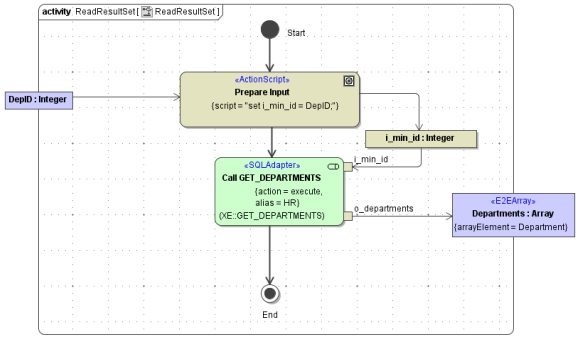
On this Page:

Related Pages:

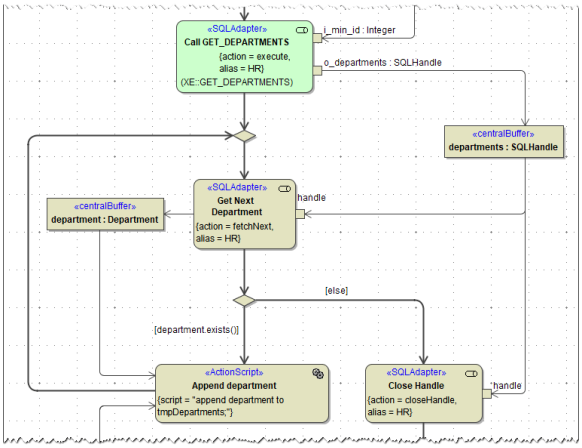
- Bulk Fetch
- SQL Deployment

- an array of all retrieved rows, if the output object is of type **Array**
- an SQL handle, if the output objects is of type **SQLHandle**.

In the first case, the adapter iterates over the cursor and puts all results into the output array. In the second case, the adapter just returns the **SQLHandle** enabling the modeler to iterate over the cursor. This approach is to be preferred if the result set is big. The following two activity diagrams show these two options of calling **GET_DEPARTEMENTS**.



If the result set is big we recommend using the cursor explicitly:



You can find more information in the stored procedures example.

> You cannot run the example on the SQLite databases we deliver with the Bridge tool set. If you have an Oracle database you could set up the stored procedures there and change the component diagram in the model to run it against your database.

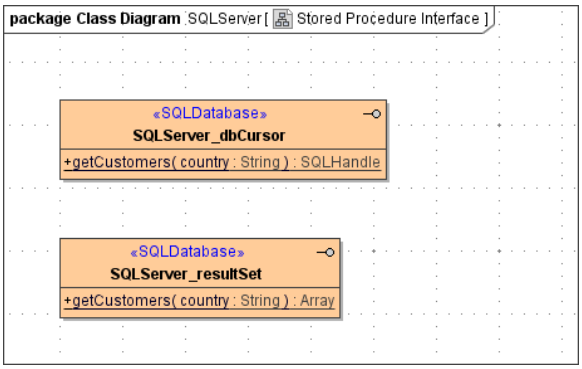
Handling Stored Procedures Returning a Result Set

Added in Builder 6.0.15.1 Requires Runtime 2015.15 Some databases support stored procedures returning a result set. Handing this output is very similar to Handling Stored Procedure Output Containing Multiple Records.

For example, on an SQL Server such a stored procedure may look like:

```
create procedure getCustomers
  @country VARCHAR(100)
as
begin
  SELECT [CUSTNO] ,[COUNTRY] ,[NAME1] ,[NAME2]
  FROM [E2E_Examples].[dbo].[customer]
  WHERE [COUNTRY] = @country
end;
```

To receive the result set, define a parameter with **direction return** in your model on the related stored procedure and apply **stereotype** <<SQLReturnResultSet>> to this parameter.



Depending on the **UML** output type of the return parameter, the SQL adapter will return

- an array of all retrieved rows, if the return parameter is of type **Array**
- an SQL handle, if the return parameter is of type **SQLHandle**.

In the first case, the adapter iterates over the cursor and puts all results into the output array. In the second case, the adapter just returns the **SQLHandle** enabling the modeler to iterate over the cursor. This approach is to be preferred if the result set is big.

> You can name the parameter as you like. The name doesn't have to be *result* (as it is in the example) – only criteria are the direction, the stereoptype and the datatype.