HOW TO turn a database into a LinkableComponent

HOW TO create a database reader

In the same way as for the ASCII reader, you can expose data contained in a database via the OpenMI by replacing the methods that implement the ExchangeItems and the GetValues calls. Furthermore it is possible to store data in the database through the SetValues call. One way of implementing this is to use SQL statements.

Accessing databases

In the OpenMI, the delivering component is obliged to return a set of values for the requested time. Each GetValues request initiates a processing activity. As many water-related models progress in time, the time argument is the controlling variable for any processing activity. Linkable components that do not progress in time can neglect the time argument; they do the required processing and return the data. However, they must be able to pass the time argument to another component if they invoke a GetValues call themselves.

If the requested time does not match the timestep in the computation and the computation is already ahead, an interpolated value must be delivered. If the computation has not yet reached the requested time, two alternatives are available:

- Calculate the values at the requested time.
- Extrapolate the solution.

Under normal conditions the first alternative should be chosen. For bidirectional links, one of the components will need to extrapolate its solution in order to prevent deadlock situations.
Components that do not progress in time, such as databases, should also be able to return a value, whether it is the actual, interpolated or extrapolated one. For those situations where the exact time information is required, an additional interface is provided to obtain the discrete timestamps available. Implementation of this interface, IDiscreteTimes, is optional.

Accessing databases using ADO.NET

The .NET environment offers a set of classes to access different kinds of database (MS-Access, SQL-Server, Oracle). Developing a tool that saves data from a GetValues call to a database is not difficult. The steps are as follows:

- Connect to an SQL server (Figure 1).

```
// C#
    this.sqlConnection1 = new System.Data.SqlClient.SqlConnection();
    this.sqlConnection1.ConnectionString = "data source=riz02\\NETSDK; " +
        "initial catalog=OpenMI;" +
        "user id=User;password=Open;" +
        "persist security info=True; " +
        "workstation id=riz02; " +
        "packet size=4096";
```

Figure 1 Connecting to an SQL server

- Execute an SQL command (Figure 2).

```
//
// SQL command Select
//
this.sqlSelectCommand.CommandText = " select * from measurement";
this.sqlSelectCommand.Connection=this.sqlConnection1;
//
// SQL command Insert
//
this.sqlInsertCommand.CommandText= " insert into OpenMI.measurement
                              (Quantity, Time, Value)
                        VALUES (QuantityID, Timestamp, Values[0])
this.sqlSelectCommand.Connection=this.sqlConnection1;
```

Figure 2 Executing an SQL Command

This is an example of accessing an SQL-server database. The .NET environment provides classes to access many types of database, including MS-Access.