

Use of HiLoGenerator with sql-util DAO classes

Add table ID_GENERATOR to TableDefinitions

```
TableInfo idGenerator=new TableInfo("ID_GENERATOR");
idGenerator.setColumns(Arrays.asList(
    new ColumnInfo("NEXT_ID",Types.BIGINT,20,null,false,null)
));
```

Future versions should use the constants `HiLoIdGenerator.DEFAULT_ID_TABLE_NAME`, `HiLoIdGenerator.DEFAULT_NEXT_ID_COL_NAME` which are part of sql-util as of the yet unreleased version 1.2.1.

Instantiate an idGenerator bean

```
<bp:bean id="idGenerator" class="org.clazzes.util.sql.dao.HiLoIdGenerator"
    init-method="initialize" depends-on="databaseSetup">
    <bp:property name="dataSource" ref="dataSource"/>
</bp:bean>
```

databaseSetup is the bean representing the JDBC2XML SchemaManager (see [this HowTo](#)), depends-on assures, that the ID generator is started after the database tables are set up.

Variant: idGeneratorbean using application specific table and column names:

```
<bp:bean id="idGenerator" class="org.clazzes.util.sql.dao.HiLoIdGenerator"
    init-method="initialize" depends-on="databaseSetup">
    <bp:property name="dataSource" ref="dataSource"/>
    <!-- idTableName, default: ID_GENERATOR -->
    <bp:property name="idTableName" value="MYAPP_ID_GENERATOR"/>
    <!-- nextIdColumnName, default: NEXT_ID -->
    <bp:property name="nextIdColumnName" value="NEXT_ID"/>
</bp:bean>
```

Set the sqlGenerator in your DAOs

```
<bp:bean id="userDAO" class="at.egv.drust.impl.dao.jdbc.DsUserJdbcDAO">
    <bp:property name="idGenerator" ref="idGenerator"/>
    <bp:property name="threadLocalKey" ref="jdbcUrl"/>
</bp:bean>
```