jdbc4olap dev guide

Environment setup

Eclipse

The development of jdbc4olap was made with Eclipse 3.3.0. Other versions should be fine with this guide as long as they support the different plugins. You also need a JDK 1.5 to run the project.

Subclipse

You need the Subversion plugin for Eclipse to deploy the project. You'll find it and detailed info for installation here http://subclipse.tigris.org.

Once installed you can checkout the project from the following repository:

```
https://jdbc4olap.svn.sourceforge.net/svnroot/jdbc4olap/trunk
```

JavaCC

You also need the JavaCC Eclipse plugin, available here http://sourceforge.net/projects/eclipse-javacc, to precompile the parser. Right click/compile with JavaCC on sql.jjt from org.jdbc4olap.parsing will generate all required classes.

Architecture of jdbc4olap

packages

jdbc4olap is composed of 3 packages.

- org.jdbc4olap.jdbc: contains the classes that implement JDBC specification:
  - OlapDriver: where it all begins, registers the driver, get the server's properties and attempts to connect if the URL provided is accepted.
  - OlapConnection: that's the heart of the driver. Makes connections to olap servers and initiates MetaData and Statement objects. A connection is in fact a dialog attempt with the OLAP server, by sending a XMLA message and checking the validity of the response. Indeed, XMLA is used in stateless mode, so that most providers are supported.
  - OlapDatabaseMetadata: used to discover a database, by querrying catalogs, schemas, tables, columns, imported keys and primary keys.
  - OlapPreparedStatement, OlapStatement: used to launch SQL querries. The SQL-MDX conversion and result extraction and layout are made here, thus it's probably the most critical part of the code.
  - OlapColumnMetadata, OlapResultSet, OlapResultSetMetaData: used to encapsulate data.
- org.jdbc4olap.parsing: provided with 2 files:
  - sql.jjt: grammar script for parsing an SQL expression and generating a tree representation of that expression.
  - SimpleNode: modified version of a generated class. Allows to attach some text to a node.
- org.jdbc4olap.xmla: classes that handle XMLA manipulation:
  - XmlaConn: used by OlapConnection, creates XMLA messages and extract relevant parts of the replies.
  - PropertyManager, StandardPropertyManager: interface and implementation of a XMLA properties manager. Retrieves properties, identify the read-only from the writable ones, the JDBC-mapped ones, and the deprecated ones.
  - XmlaProperties, XmlaRestrictions: manage the two different parameters of a XMLA message.
  - QueryColumn, QueryFilter, QueryFilterOperand, QueryTable, XmlaTuple: used to extract and manipulate the result of a query.