DB Browser
Technical architecture

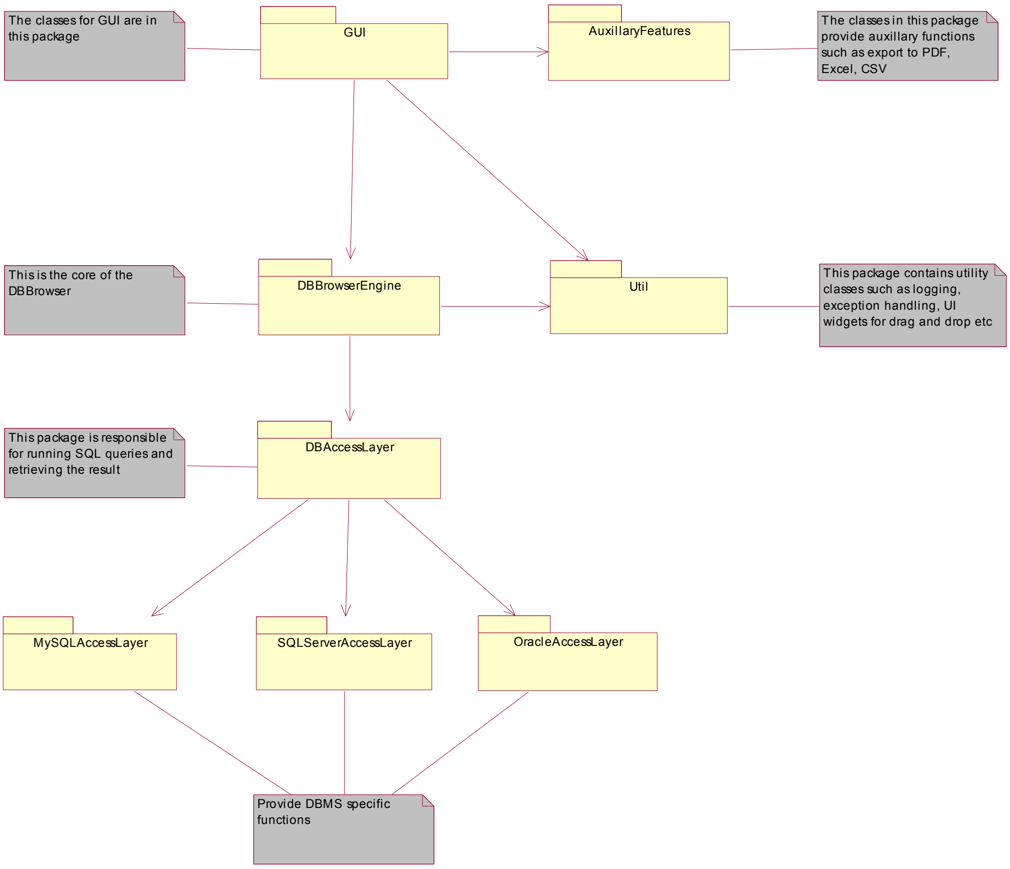
Contents

Introduction

DBBrowser is an open source, cross-platform tool which can be used to view the contents of a database. It supports CLOBS, BLOBS and Oracle XMLTypes. It is designed to work with all the major DBMS (Oracle, MySQL, SQLServer). The user should never have to write SQL to view the data although a SQL window is provided. Support for ER (Entity Relationship) diagrams is planned for the next version.

DBBrowser is hosted on the SourceForge website (http://sourceforge.net/projects/databasebrowser).

Architecture


The classes for GUI are in this package
GUI
AuxillaryFeatures
The classes in this package provide auxillary functions such as export to PDF, Excel, CSV
This is the core of the DBBrowser
DBBrowserEngine
Util
This package contains utility classes such as logging, exception handling, UI widgets for drag and drop etc
This package is responsible for running SQL queries and retrieving the result
DBAccessLayer
MySQLAccessLayer
SQLServerAccessLayer
OracleAccessLayer
Provide DBMS specific functions

DBBrowser is divided into 3 layers.

- DB access layer
- DBBrowser engine
- User interface

JDBC Drivers

The DBBrowser should work with any DBMS for which there are JDBC drivers available. The DBBrowser does not include ant JDBC drivers and the user must download the appropriate JDBC driver and 'register' it with DBBrowser. The registration can be done through the UI. During registration, the JDBC driver classes are added to the classpath.

Connection info

Information about a connection to a Database is stored as an XML file in DBExplorer format. A sample XML connection info file is shown below:

```
<DBConnections>
  <ConnectionDef>
    <Enabled>true</Enabled>
    <Name>KFI_amangat</Name>
    <DriverClass>oracle.jdbc.OracleDriver</DriverClass>
    <Url>jdbc:oracle:thin:@svr-oracle-1:1521:devrdf03</Url>
    <User>amangat</User>
    <Password>amangat</Password>
    <SqlTerminator>;</SqlTerminator>
</ConnectionDef>
.
.
.
</DBConnections>
```

User Interface

The user interface is implemented using Swing. The reasons for choosing Swing are:

- It is well known and well understood by the Java community
- It is very flexible and can be used to develop complex user interfaces
- It integrates well with SkinLF package from SkinLF.com and allows us to change the look and feel of the application on the fly. There are several 'themepacks' available to allow the user to customise the look and feel of the application

Other options are SWT package from IBM. However it is more windows centric and not well known in the Java community but is faster than Swing

Accessing Help

It is the aim of the application developers to make the system intuitive to use. However a user guide is provided and is accessible through the application menu or via context sensitive help. The user can right click on a UI widget and access context sensitive help.

The user guide is in the JavaHelp format.

Internationalisation

Support for internationalisation is implemented through standard Java Resource bundles. DBBrowser includes resource bundle for English only.

External libraries

External libraries used in the project are:

- JDOM including Xerces– used for reading the connection info XML file (An alternative is to use higher level API such as Castor or XMLBeans)
- SkinLF – used to change the look and feel of Swing applications on the fly
- iText – used to output the result of a SQL query as PDF document
- POI/HSSF libraries from Apache Jakarta project – used to output the result of SQL query as Excel file

Export results

The contents of a table can be exported as:

- XML
- CSV – Comma Separated Values
- Excel files
- PDF documents