Kexi - Data(base) Environment
Universal Data Layer for KDE
Introducing the new codename:

Predicate

Target: KDE 4.1 or 4.2
Overview

• Problems to address
• Introduction to Predicate
• Predicate and KDE, Qt
• Kexi and Predicate plans
Problems you can solve with Predicate

- Opening depends on document size even if you want to add one character
- Saving can take even longer
Predicate – what's in the box?

- SQLite, MySQL and PostgreSQL support
- Cost of data opening and saving - independent of total size (unlike the case with big XML files)
Database connectivity …for human beings?

• It should take a minute to start working with new data…
Startup assistants to the rescue
Unified database connection files

[Connection1]
caption=Tunnel to the main server
encryptedPassword=**************
engine=mysql
port=3333
type=connection
useLocalSocketFile=false
user=test

- Admin defines server connections
- Users refer to them by name from any app
“Database import/export” assistants

![Database Importing window](image)
Database connectivity for ...human beings?

• Want more users?

Do not expect database administration skills.
Problems you can solve with Predicate (2)

- Don't export, don't import. Share. (If possible)
- Document files come from the pre-network and pre-Internet era
- Avoid creating multiple versions of the same document
Database connectivity for ... human beings?

- The solution: local database files with ability of migrating to a server
- Implementation: SQLite 3
• Self-contained C library acting as SQL server for application it links to

• Features:
  • Transactions
  • Fast, constant access time (binary format)
  • Command line interface
  • Widely adopted
• SQLite in KDE:
  • Default Kexi (and thus, Predicate) database format
  • Redland RDF's storage used by Soprano, which is used by NEPOMUK-KDE
  • KDE PIM's Akonadi file-based storage
  • Amarok's music collection
  • Apps using QtSql can already use SQLite
in other software

- File format of Adobe Lightroom and Adobe Integrated Runtime
- Unified storage backend for Mozilla 2 components and extensions
- Included in Python, PHP and RealBasic
- Used by Google Desktop for the Mac
- Storage for AJAX apps in Google Gears
SQLite as system component in OS-es

• Part of Mac OS X 10.4 as embedded db engine; also used in Apple Mail and Aperture
• Integral part of Symbian 9.5
• Service Management Facility replacing /etc/inittab in Solaris 10
Can Predicate work for my app?

- Many applications are internally already database-like
  - especially those employing MVC design pattern
- There's no need to deal with database internals (SQL, backend specifics)
By the way: Predicate and KIO

- Predicate is like KIO interface but for database and data structures
<table>
<thead>
<tr>
<th>KDE Interface:</th>
<th>KIO</th>
<th>Predicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data model</td>
<td>Hierarchical (tree/directed graph)</td>
<td>Relational, possible object-relational and hierarchical extensions</td>
</tr>
<tr>
<td>Basic concepts</td>
<td>inode (file/directory)</td>
<td>Record, column, table, relation</td>
</tr>
<tr>
<td>Relation between objects</td>
<td>one parent-many children/many-many (links)</td>
<td>record-in-table/view, relationships between records (graph)</td>
</tr>
<tr>
<td>Types and properties</td>
<td>File, directory (file types not handled at this level), file/directory attributes</td>
<td>Column type (simple data types)</td>
</tr>
<tr>
<td>Permissions</td>
<td>At file level, optionally for part of file</td>
<td>Up to individual cell, by controlling access to tables, records and columns</td>
</tr>
<tr>
<td>Namespaces</td>
<td>URI scheme name, mount points, elements of path</td>
<td>Data source/schema/table or view name/column name/alias</td>
</tr>
<tr>
<td>Generalized Family of abstracted</td>
<td>Anything supporting URI schemas: filesystem/URL-based network protocol (app. layer)/virtual filesystems</td>
<td>SQL databases/tabular data sources (CSV, XML)</td>
</tr>
</tbody>
</table>
Predicate – what's in the box? (2)

- Tools for creating data models are available at API level – no need to hardcode SQL statements and implement GUI

```java
TableSchema t = new TableSchema("ownership");
Field f = new Field("id", Integer);
t.addField(f);
f.setPrimaryKey(true);

f = new Field("owner", Integer);
t.addField(f);

f = new Field("model", Integer);
t.addField(f);

f = new Field("since", Integer);
t.setLookupFieldSchema("owner",
```
Predicate – what's in the box?

- Built-in SQL parser provides probably the only Qt-based framework for employing type-safe SQL queries within applications.

![SQL Query Text]

- The query is incorrect

Table "cars" is covered by aliases. Instead of "cars.id", you can write "c.id"
Predicate – what's in the box? (3)

- **Strong data typing and constraints** (general feature of databases)
  - Types and support for constraints generalized for typical usage

<table>
<thead>
<tr>
<th>Id</th>
<th>Place</th>
<th>Date</th>
<th>Photo</th>
<th>Privacy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Michał</td>
<td>2006-03-22</td>
<td><img src="image" alt="Photo" /></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Augustów</td>
<td>2006-06-20</td>
<td><img src="image" alt="Photo" /></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>OpenSUSE</td>
<td>2006-12-20</td>
<td><img src="image" alt="Photo" /></td>
<td></td>
</tr>
</tbody>
</table>

Error: Date?
# Predicate and QtSql

<table>
<thead>
<tr>
<th>API features</th>
<th>QtSql</th>
<th>Predicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Definition Language</td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Data Manipulation Language</td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>SQL</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>SQL Parsing</td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>Data &amp; Project Migration</td>
<td></td>
<td>+</td>
</tr>
<tr>
<td>MVC API</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
Predicate and ODBC

- ODBC specification is good but its implementation is simplistic (you may need to depend on 3rd party commercial drivers)
- ODBC is not a solution as your appetite for generic data API increases
- So there's a plan for Predicate to wrap ODBC
Predicate – what's in the box? (4)

• MS Access data import
• Database import from PostgreSQL and MySQL servers to a file
Predicate – what's not included?

• Database administering and tuning tools (use dedicated for particular db backend)

MySQL Administrator

pgAdmin PostgreSQL Tools
Predicate – how to get it?

• KDE SVN: trunk/koffice/kexi/kexidb/

• will be moved to trunk/???/predicate/
Predicate – Plans

• Providing Kexi plugins for other KDE apps: designer tools for tables, queries and forms, data widget, simple printouts

• Macros framework (in MS Access style) - a good companion for Kross
Predicate – Plans

• Providing Kexi plugins for other KDE apps:
  • table, query and form designer tools
  • data-aware widgets
  • simple printouts
  • CSV import/export
Predicate – Plans

• A way for dealing with native data sources and queries: deal with them as black boxes not available for direct data binding to the GUI, etc.

• Advantage: proprietary or not yet supported features can be used out of the box by knowledgeable users

• Side effects: executing native DML queries can introduce inconsistencies
Kexi 2 - Plans

- More SQL features (grouping, improving the parser, alter table)
- Full text search, bindings to the upcoming KDE search facilities
- Reports using KOffice text layout engine with support for ODF
Kexi 2 – Plans (2)

• Multithreading – optimization for large data sets

• Live data sharing ala (MS DDE) Dynamic Data Exchange, hopefully a companion to D-BUS

• Think about Qt-only and/or non-Qt users of the facilities?
Thank you.
Questions? Ideas?

More info:
- http://www.kexi-project.org
- http://www.koffice.org/kexi

API Docs:
- http://www.kexi-project.org/predicate