ODBMS.ORG www.odbms.org

Editor: Roberto V. Zicari

TechView Product Report: db4o

Nov. 7, 2008, Revised June 2010.

Product Name: **db4o**
Company: **Versant**
Respondent Name: **Carl Rosenberger,**
                    Chief Software Architect db4o, Versant.

1. Support of Programming Languages

How is your product enabling Java. .NET/C#. C++, developers to store and retrieve application objects? What other programming languages do you support?

db4o stores any Java or .NET object with one line of code:

```
ObjectContainer#store(object);
```

When an object is stored, all attached objects are stored as well (persistence by reachability). db4o also supports transparent persistence.

When objects are loaded from a database, references between objects are restored. db4o ensures that each persistent object is loaded only once per client.

When persistent objects are traversed, the user can decide whether he wants to take care of loading member objects explicitly (manual activation) or whether he wants members to be loaded on first access by the application (transparent activation, requires enhancing application classes).

db4o runs on Java platforms that provide a subset of Java comparable to JDK 1.1, including Android and J2ME CDC. Scala, Groovy and JRuby are also supported. db4o is also provided as a native .NET version where it works with all .NET languages that support reflection. db4o is available for .NET 2.0, 3.5 and 4.0 and for all corresponding CompactFramework implementations.

2. Queries

Does your product support querying? If yes, can you describe the querying mechanism, giving examples of queries?

db4o supports the following four different querying mechanisms:

(1) *Query-By-Example*
```
Pilot pilot = new Pilot();
pilot.setPoints(100);
ObjectSet result = db.queryByExample(pilot);
```

(2) *SODA*
```
Query query = db.query();
query.constrain(Pilot.class);
query.descend("points").constrain(new Integer(100));
ObjectSet result = query.execute();
```

(3) *Native Queries*
```
List <Pilot> result = db.query(new Predicate<Pilot>() {
    public boolean match(Pilot pilot) {
        return pilot.getPoints() = 100;
    }
});
```

(4) *LINQ (.NET only)*
```
var result = from Pilot pilot in db
    where pilot.Points == 100
    select pilot;
```

How is your query language different from SQL?

The emphasis of db4o queries is on providing 100% type safety, 100% refactorability and 100% compile-time checks. This goal has been achieved with three out of the four querying mechanisms: Queries are written in the respective programming language and not as embedded strings (Only exception: SODA needs strings for field names.)
By not using embedded strings developers can be more productive and the quality of the resulting code can be higher.

Does your query processor support index on large collections?

No.

3. Data Management

Do you provide special support for large collections?

Yes. We have a BigSet class based on BTrees.

How do you handle data replication?

db4objects provides a separate product "dRS" to replicate objects between db4o databases, to Versant VOD object databases and also to relational databases via Hibernate.

How do you handle data distribution?

db4o provides a Client/Server mode, where each client can have it's own cache of persistent objects.

How do you handle schema evolution?

db4o internally stores a superset of all used schemas. If fields are added to a class, they can simply be used immediately, without any required maintenance run over the database. Similarly fields can easily be removed. API calls are available for renaming fields and for retrieving field values from a previous schema where a field existed that is no longer present.

Please describe an object lifecycle when an object is loaded from a database: When are members of an object loaded into memory? When are they discarded?

In manual activation mode, db4o uses a "depth" concept: When an object is returned from a query, members are loaded up to a certain preconfigured "depth" – the number of references away from the root object. If more objects need to be instantiated at once, db4o offers a method call

      ObjectContainer#activate(object, depth)

to do this.

In transparent activation mode, db4o tracks access to fields of objects. Whenever a field is first accessed, db4o loads the field value from the database.

db4o supplies special collections for Java and for .NET that support transparent activation and transparent persistence. The db4o enhancer tool can be run at deployment time to automatically exchange collections in user applications to use persistence-aware collections.

db4o holds weak references to instantiated persistent objects: When the garbage collector detects that there are no longer other hard references to a persistent object, it can discard the object from memory.

4. Data Modeling

Which Data Modeling notation do you recommend for the design of your data?

db4o detects the schema directly from application classes. Any available tool can be used to model these. We recommend the use of Eclipse for Java and Visual Studio for .NET languages.

5. Integration with relational data

Could you integrate flat relational tables into your object database? If yes, how?

It is possible to create a class and treat it as a relational table where the fields of the class represent relational columns.

6. Transactions

How do you define a transaction? Pls. use a simple example.

Work with db4o always is transactional. Commiting a transaction or rolling it back immediately starts a new transaction.
The corresponding API methods are:

ObjectContainer#commit()

and

 ObjectContainer#rollback().

Is there a way to discard objects that have been modified in the current transaction from memory?

Yes. The corresponding API method calls would be:

ObjectContainer#store(object)

ObjectContainer#purge(object)

After a call to ObjectContainer#store() *garbage collection will discard objects that are no longer reachable in the application. Modifications to objects will still be committed with the transaction.*

Do you provide a mechanism for objects on clients to stay in synch with the committed state on the server? If yes, please describe how it works.

Yes. We call our mechanism "Pushed Updates". It is possible to register listeners on clients to listen to all changes that are committed by other transactions. In the listener it is possible to refresh the persistent object with the current committed state.

How are transaction demarcations for your product expressed in code?

Explicit API calls are required:

ObjectContainer#commit(),

ObjectContainer#rollback().

Describe the strategies possible with your product for concurrent modifications by multiple clients.

We supply a semaphore system and recommend to use it to guard code blocks from concurrent execution.

To keep objects on clients in sync with committed state we recommend to use our "pushed-update" functionality.

7.Persistence

Are there requirements for classes to be made persistent? If yes, please describe them.

There are absolutely no requirements. An object of any class can be stored as is, if manual activation and manual persistence is used.

Does your product require enhancement of application classes for Transparent Persistence? If yes, briefly describe the additional steps required for a build process of an application

In order for Transparent Persistence and Transparent Activation to work, classes need to support our Activatable interface. This can either be done by implementing the interface by hand or by enhancing classes with our enhancer tools.

The recommended process for Java is to add an enhancer Ant script to the Eclipse project as an Ant builder. It is also possible to enhance classes at class-loading time by using a special ClassLoader.

On .NET we recommend to call the db4o enhancer tool "Db4oTool.exe" as a post build event from Visual Studio.

8. Storage

Does your product operate against a single database file or are multiple files required?

db4o runs against a single database file. It is very easy to set up databases. One line of code to open the database is sufficient. If a database does not exist it will be created.

9. Architecture

Does your product support multiple clients connecting to a single database?

Yes.

For which Operating Systems is your product running?

Since db4o is written in Java and in .NET, all operating systems that support Java or .NET will also run db4o.

db4o for Java also runs on many limited Java dialects including Android and J2ME CDC.

db4o for .NET also runs on CompactFramework (PocketPC devices) and on Silverlight..

10. Application

What kind of applications are best supported by your product?

db4o is not limited to any particular application. We see happy users of db4o working on devices with limited resources and we also see them using db4o as a cache in near-real time systems.

Object databases play out their advantages best in applications with complex object graphs. Thorough object-oriented design works well with object databases.

11. Performance

What benchmark do you use to measure the performance of your system?

We like to work with the Poleposition benchmark, http://www.polepos.org.