

New db4o Replication System v8 in a Nutshell

by Carl Rosenberger <carl@db4o.com>, db4o Chief Software Architect, Versant Corporation

A new improved version of the db4o Replication System (dRS) is available as part of db4o's general db4o 8.0 production release. dRS now also supports replicating objects with the Versant Object Database (VOD).

dRS allows to keep track of changes to multiple databases, copy modified objects between them and thus keep their state in sync. Use cases for replication include distribution of data to disconnected devices, migration, disconnected WAN networks, failover security, incremental backups and more.

dRS allows querying for objects that were added, deleted or modified since the last time two databases were replicated against each other. Copying object state across databases is done with a single *#replicate()* API call.

```
// Start a replication-session
ReplicationSession replicationSession =
    Replication.begin(mobileDatabase, centralDatabase);

// Get all changes since the last replication
ObjectSet changedObjects =
mobileDatabase.objectsChangedSinceLastReplication();

// Replicate all changed objects
for (Object changedObject : changedObjects) {
    replicationSession.replicate(changedObject);
}
// Commit the replication session
replicationSession.commit();
```

Unique: Object-oriented Synchronization Conflict Resolution

One of the unique features of the dRS is its database (rather than application) driven conflict resolution. The application data model, which should be the sole repository for data-oriented business logic, can be used to solve conflict resolution, rather than relying on secondary application code repositories.

Synchronization conflicts occur when an object is simultaneously modified in disconnected database instances since their last replication session. The user may then choose to model business rules that automatically resolve these conflicts

by overwriting changes from one side to another, skipping the conflicted object, stopping replication to avoid data corruption and/or embedding a notification routine to trigger further user action or other business rules.

```
ReplicationEventListener listener = new ReplicationEventListener() {
    public void onReplicate(ReplicationEvent event) {
        if (event.isConflict()) {
            //Override the record with PDA's copy (Provider A)
            event.overrideWith(event.stateInProviderA());
        }
    }
};
```

For further details about the dRS API please see the [reference manual \(http://developer.db4o.com/Documentation/Reference/db4o-8.0/java/reference/\)](http://developer.db4o.com/Documentation/Reference/db4o-8.0/java/reference/).

New: Replication with Versant Object Database (VOD)

The most prominent new feature in the new dRS release is support for replication against VOD. To keep track of changes in VOD, dRS relies on the Versant event system and on installing the dRS EventProcessor that is shipped with dRS. To demonstrate the setup we provide a dedicated "drsVodExample" Java project in our [SVN repository \(https://source.db4o.com/db4o/trunk/sandbox/drsVodExample/\)](https://source.db4o.com/db4o/trunk/sandbox/drsVodExample/).

In a nutshell setting up db4o to VOD replication requires the following steps:

- Ensure you have VOD 8.0.1.3 or later installed. (Download from [here. http://www.versant.com/developer/downloads/](http://www.versant.com/developer/downloads/)) Make sure you select JVI support in the installation wizard. It's a subitem beneath 'Java'.
- Create the event schema in the database you want to replicate with.
- Make sure the Versant event daemon and the dRS event processor are running.

Getting the drsVodExample project set up on your machine will help you to find out how all of the above works. For details on the individual steps please refer to the README.txt file in the drsVodExample project and to the dRS reference documentation.

Other modifications to dRS include the following:

- The timestamp mechanism has been improved to provide better support for concurrent modifications while replication is running. Previously db4o

used a version timestamp that was generated when an object was written in a call to *#store()*. With this approach concurrent modifications may have been accidentally included a running replication session. The old mechanism has been replaced by a true commit timestamp. With the new approach concurrent modifications are deterministically excluded from the running replication session. They can be replicated in the next replication session.

- Object versions ('Versions') in db4o have been ported to global BTree-based indices ('Commit Timestamps'). As a result the lookup performance for versions has been improved by an order of magnitude.
- The following APIs change in db4o from version 7.12 to version 8.0:
 - *FileConfiguration#generateVersionNumbers()* becomes *FileConfiguration#generateCommitTimestamps()*
 - *ObjectInfo#getVersion()* becomes *ObjectInfo#getCommitTimestamp()*
- Migration from db4o 7.12 to 8.0 converts old version numbers to new commit timestamps automatically when a database is first opened with db4o 8.0.

dRS is available for replicating objects between db4o, Versant Object Database (VOD) and relational databases through Hibernate.

dRS 8.0 can be downloaded from here (<http://developer.db4o.com/Downloads.aspx>)