

Versant db4o

Présentation db4o

- Le projet db4o a démarré en l'an 2000.
- La société Db4objects Inc a été créée en 2004 pour commercialiser le produit
- Db4o a été acheté par Versant en 2008 (Open Source)
- L'échange de données entre db4o et une base de données relationnelle peut être réalisé à l'aide de db4o Replication System (dRS).

Présentation db4o

- Téléchargement
 - db4o for java
 - db4o for .Net 4.0
- Deux modes d'utilisation
 - Local ou embarqué
 - Client serveur
- Taille maximale de la base de données: par défaut 2Go (Taille d'un bloc 1 octet), et peut atteindre 254 Gb (Taille d'un bloc 127 octets)
- Taille recommandée 16 (Taille d'un bloc: 8 octets qui est aussi la taille des pointeurs en interne)

- Performances
 - nombre de champs par objet > 1000
 - Environnement concurrent : 10 utilisateurs /s
 - requêtes: 200000 objets/s
- Le moteur de base de données consiste en un seul fichier jar: db4o-8.0.249.16098-all-java5 (il faut l'ajouter dans le classpath).

Outils

- OME (Object Manager Enterprise):
explorateur d'objet (plugin eclipse)
 - Plugin: ome/ObjectManagerEnterprise-Java-8.0.0.zip dans un dossier.

Persistance des objets

Package com.db4o

Ouvrir la base de données

```
ObjectContainer db = Db4oEmbedded.openFile("base1.yap");
```

 si la base données n'existe pas, elle sera créée.

Ajout d'un objet

```
db.store(c);
```

Rechercher un objet

db4o fournit plusieurs systèmes d'interrogation de données: QBE (Query By Example), NQ (Native Query), SODA (Simple Object Database Access) et LINQ (Pour dotnet)

Recherche par une requête QBE

1. Créer le prototype à utiliser pour la recherche:

```
Client c1 = new Client();  
c1.setId( 2);
```

2. Rechercher les objets

La méthode db.queryByExample retourne tous les objets de type Client qui ont les mêmes que celles de l'objet c1 (les valeurs par défaut sont ignorées), les objets sont retournés dans une collection de type IObjectSet

```
ObjectSet<Client> listeClients = db.QueryByExample(c1);  
Client c1= listeClients.next();
```

Fermer la base de données

```
db.close();
```

Suppression

recherche de l'objet + db.delete(objet)

Rechercher tous les objets d'un même type

```
Client c1 = new Client();  
ObjectSet<Client> clients = db.queryByExample(c1);
```

Ou

```
ObjectSet<Client> clients =  
    db.queryByExample(Client.class);
```

Modifier un Objet

Rechercher d'abord d'abord l'objet à modifier

```
Client proto= new Client();  
proto.setId(2);
```

```
ObjectSet<Client> listeClients =  
    db.queryByExample(proto);  
    Client c1= listeClients.next();  
c1.setId(5);  
    db.store(c1);
```

Objets complexes

Mise à jour

```
db.ext().Configure().objectClass(Client.class).CascadeOnUpdate(true);
```

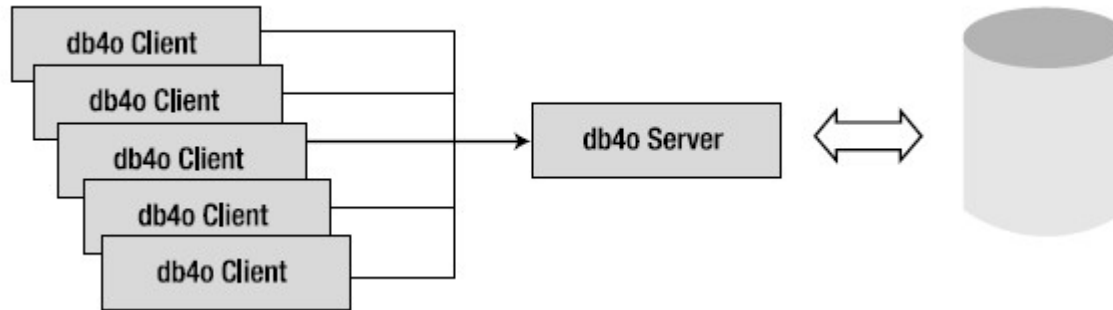
Ou

```
db.ext().configure().updateDepth(15);
```

Suppression en cascade

```
db.ext().Configure().objectClass(Client.class).CascadeOnDelete(true);
```

Mode client Serveur



- 3 modes C/S:
 - Le mode TCP/IP (Réseau)
 - Le mode embarqué
 - Le mode d'administration (sauvegarde, défragmentation, arrêt)
- Serveur
 - Démarrage du serveur
 - Connexion distante (grantAccess)

Le serveur

```
import com.db4o.ObjectServer;
import com.db4o.cs.Db4oClientServer;
...
ObjectServer db4oServer = null;
    try {
        int port = 5555;
        System.out.println("Démarrage...");
        db4oServer = Db4oClientServer.openServer("base.yap",
port);
db4oServer.grantAccess("user1", "pass1");
        System.out.println("Le serveur écoute sur le port
numéro:" + port);
        while (true) ;
    } finally {
        db4oServer.close();
    }
}
```

Le client

```
ObjectContainer db =
Db4oClientServer.openClient("localhost",
5555, "user1", "pass1");
ObjectSet<Client> personnes=

db.queryByExample(Client.class);

System.out.println(personnes.size());
    for (Personne p:personnes){
        System.out.println("Nom:" +
p.getNom());
    }
    db.close();
```

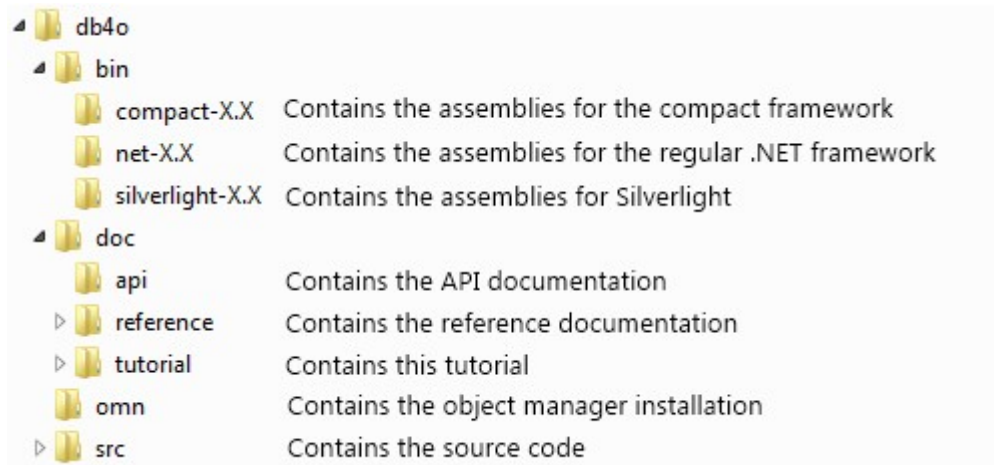
Les transactions

ACID

- Atomicité: traiter un groupe d'opérations comme une seule opération.
- Cohérence: toute transaction amène un système d'un état valide à un autre état valide (contraintes d'intégrité, rollbacks en cascade, déclencheurs...)
- Isolation: lorsque plusieurs clients accèdent à une base de données simultanément. Il est important d'isoler un utilisateur à partir des changements qui sont effectués aux mêmes données par l'autre utilisateur.
- Durabilité : est la capacité de la base de données pour récupérer les données dans le cas d'une défaillance matérielle ou de logiciel.

- Début de la transaction : `OpenFile`,
`OpenClient`.
- Validation de la transaction `db.Close()` ou
`db.Commit()`

Db4o (Dotnet)



- Le moteur de base de données consiste en un seul fichier: Db4objects.Db4o.dll (il faut l'ajouter comme référence dans tout projet .Net utilisation l'API db4o)

Persistance des objets

```
using com.db4o.ObjectServer;  
using com.db4o.cs.Db4oClientServer;
```

Ouvrir la base de données

```
IObjectContainer db=Db4oEmbedded.OpenFile("base1.yap"); si la base données n'existe pas, elle sera créée.
```

Ajout d'un objet

```
db.Store(c);
```

Rechercher un objet

db4o fournit plusieurs systèmes d'interrogation de données: QBE (Query By Example), NQ (Native Query), SODA (Simple Object Database Access) et LINQ

Recherche par une requête QBE

1. Créer le prototype à utiliser pour la recherche:

```
Client c1 = new Client();  
c1.Id = 2;
```

2. Rechercher les objets

La méthode db.QueryByExample retourne tous les objets de type Client qui ont les mêmes que celles de l'objet c1 (les valeurs par défaut sont ignorées), les objets sont retournés dans une collection de type IObjectSet

```
IObjectSet listeClients = db.QueryByExample(c1);  
c1= listeClients.Next() as Client;  
db.Store(c);
```

Suppression

recherche de l'objet + db.delete(objet)

Rechercher tous les objets d'un même type

```
Client c1 = new Client();  
IObjectSet clients = db.QueryByExample(c1);
```

Ou

```
IObjectSet clients =  
    db.QueryByExample(typeof(Client));
```

Ou

```
IList<Client> clients  
    =db.Query<Client>(typeof(Client)) ;
```

Modifier un Objet

Rechercher d'abord d'abord l'objet à modifier

```
Client c1 = new Client();  
    c1.Id = 2;  
    db.Store(c1);
```

```
IObjectSet listeClients = db.QueryByExample(c11);  
    c1= listeClients.Next() as Client;  
    c1.Nom="Client7";
```

Les requêtes LINQ

- Pour utiliser les LINQ, il faut:
 - Importer la bibliothèque Dbo4objects.Db4o.Linq
 - Ajouter les directives

```
using System.Linq;
```

```
using Dbo4objects.Db4o.Linq;
```

- Exemple de requête:

```
IEnumerable<Client> liste = from Client c in db  
                           where c.Commandes.Count() > 1  
                           select c;  
IList<Client> clients = liste.ToList();
```

Exemple

