UpgradeTo11EmbedNotes

Notes on Xindice 1.0 to 1.1 Upgrade for Embedded Use

– by Murray Altheim

These are my notes. This afternoon I downloaded the latest CVS of Xindice 1.1 and upgraded from 1.0 in client-server to 1.1 embedded. The entire process (not including download time) was about an hour (with help from Vadim to uncover elements of my boneheadedness).

Note that my particular use of Xindice 1.0 is fairly minimal, i.e., I basically open the database, store and retrieve XML documents, with most of the database functionality developed in working with my own API (called XNode). I don't currently use querying or indexing. Yet. The recent messages on xindice-dev about Lucene are quite interesting, and I'm looking forward to seeing what develops in this area.

Okay, here's what I've done:

1. Downloaded the latest CVS. Instructions are available from:
   http://xml.apache.org/cvs.html#Access+the+Source+Tree
   Anonymous:

   ```
   % cvs -d :pserver:anoncvs@cvs.apache.org:/home/cvspublic
     login
     password: anoncvs

   % cvs -d :pserver:anoncvs@cvs.apache.org:/home/cvspublic co xml-xindice
   ```

   Non-anonymous:

   ```
   % cvs -d YOUR_USER_NAME@cvs.apache.org:/home/cvs co xml-xindice
     login
     password: [password]
   ```

2. I built the source:

   ```
   % ant build
   % ant jar-release
   ```

3. Since that worked, I substituted the new xindice 1.1 directory for the existing xindice 1.0 directory within my project project directory (i.e., xindice is a subdirectory of my project directory). The ant build.xml file for my project reads the jar files found in the $XINDICE_HOME/java/lib directory already, so no changes were necessary here. (I've also got an eXist implementation within, and choose between them via a property.) I moved the xindice-1.1b4-dev.jar file from the dist directory to the $XINDICE_HOME/java/lib directory, and renamed it xindice.jar.
4. In my own application code I replaced the references as below:

   ```
   // m_driver = "org.apache.xindice.client.xmldb.DatabaseImpl";
      m_driver = "org.apache.xindice.client.xmldb.embed.DatabaseImpl";

   // m_root = "xmldb:xindice://localhost:4080/db/";
      m_root = "xmldb:xindice-embed:///db/";
   ```

5. Given the definitions in #4 above, the guts of the code to open an embedded database, set up a CollectionManagementService and DatabaseInstanceManager (whose only purpose is to shut down the database, something I probably won't do anymore now that I'm using embedded) is as follows:

```
        database = (Database)Class.forName(m_driver).newInstance();

        database.setProperty("db-home","./xindice");
        database.setProperty("managed","true");

        org.xmldb.api.DatabaseManager.registerDatabase(database);
        rootCollection = org.xmldb.api.DatabaseManager.getCollection(m_root);

        org.xmldb.api.modules.CollectionManagementService colMgmtService
                = (CollectionManagementService)rootCollection.getService(
                    "CollectionManagementService","1.0");

        org.apache.xindice.client.xmldb.services.DatabaseInstanceManager xdbInstMgrService
                = (org.apache.xindice.client.xmldb.services.DatabaseInstanceManager)
                    rootCollection.getService("DatabaseInstanceManager",
                    Command.XMLDBAPIVERSION);
```

As a side note, I had developed a rather complex listener negotiation scheme to handle situations where the application was opened without the database being active, the database server being opened via a different VM session than the client, etc. All of this code can now be dumped due to the simplicity of using the embedded drivers, though I might keep it around in case I decide to go back to the more complex world of client-server (for times when I do want to make available a remote database to a set of external clients).

After I'd debugged the small changes to my code, my application came up and the database opened up without any difficulty. I tried creating and deleting collections and nodes. When this worked, I quit, then substituted my existing, populated 1.0 'db' directory for the new directory, then reopened my application. To my surprise, my old collections and nodes still worked. (I do all my access to Xindice via the XNode API, so I can't really comment on any other features such as query, indexing, etc.)

I'd like to thank once again Vadim Gritsenko for his help in getting me up and running with 1.1. It was much less painful than I'd envisioned, in fact, it wasn't painful at all. His hard work has been invaluable in keeping this project alive, and if there were some kind of online applause button, I'd be pushing it right now.

Murray