SoftwareEntwicklung ▸ Beratung ▸ Schulung

Versioning in Java

How to use SVNKit in Java Development

Web Site:
www.soebes.com
Blog:
blog.soebes.com
Email:
info@soebes.com

Dipl.Ing.(FH) Karl Heinz Marbaise

Agenda

1. Short Overview
2. The first Step
3. svn log
4. svn list
5. svn rm/add/mv/cp

1. Basics about SVNKit

- SVNKit can be used freely in Open Source applications under the TMate Open Source License.
- Commerical License for closed source applications.
- SVNKit is a native Java library which supports all Subversion operations without installation of a Subversion client.

2. The First Step Initialization

- Initalize the library to work with different protocols like:
  - http/https
  - svn/svn+ssh
  - file.

```
DAVRepositoryFactory.setup();
SVNRepositoryFactoryImpl.setup();
FSRepositoryFactory.setup();
```

- Limitation: file works only with FSFS based repositories.

2. The Second Step Authentication

```
String url = "URL of your repository";
String name = "anonymous";
String password = "anonymous";

SVNRepository repository = null;
try {
     repository = SVNRepositoryFactory
          .create (SVNURL.parseURIDecoded (url));

     ISVNAuthenticationManager authManager =
          SVNWCUtil.createDefaultAuthenticationManager(name, password);

     repository.setAuthenticationManager (authManager);

} catch (SVNException e) {
     //Error...
}
```

3. svn log

```
final ArrayList<SVNLogEntry> logEntries = new ArrayList<SVNLogEntry>();
.
.
repository.log(
      new String[] {""}, //targetPaths
      startRevision,
      endRevision,
      true, //changedPath
      true, //strictNode
      new ISVNLogEntryHandler() {
            public void handleLogEntry(SVNLogEntry logEntry) {
                  logEntries.add(logEntry);
            }
      }
);
```

3. svn log

```
for (Iterator entries = logEntries.iterator(); entries.hasNext();) {
     SVNLogEntry logEntry = (SVNLogEntry) entries.next();

     System.out.println("revision: " + logEntry.getRevision());
     System.out.println("author: " + logEntry.getAuthor());
     System.out.println("date: " + logEntry.getDate());
     System.out.println("log message: " + logEntry.getMessage());
     .....
}
```

3. svn log

```
for (Iterator entries = logEntries.iterator(); entries.hasNext();) {
....
     if (logEntry.getChangedPaths().size() > 0) {
          Set changedPathsSet = logEntry.getChangedPaths().keySet();

          for (Iterator changedPaths = changedPathsSet.iterator();
               changedPaths.hasNext();) {
               SVNLogEntryPath entryPath = (SVNLogEntryPath) logEntry
                    .getChangedPaths().get(changedPaths.next());

               System.out.println(" " + entryPath.getType() + " "+
                    entryPath.getPath()
                    + ((entryPath.getCopyPath() != null) ? " (from " +
                         entryPath.getCopyPath() + " revision " +
                         entryPath.getCopyRevision() + ")" : "")
               );
          }
     }
}
```

4. svn list

- Check to see if given URL corrospond to a file or folder.

```
...
SVNNodeKind nodeKind = repository.checkPath( "" ,  -1 );
if ( nodeKind == SVNNodeKind.NONE ) {
      System.err.println( "There is no entry");
      System.exit( 1 );
} else if ( nodeKind == SVNNodeKind.FILE ) {
      System.err.println( "A directory was expected.");
      System.exit( 1 );
}
...
```

4. svn list Helpermethod

```
public static void listEntries( SVNRepository repository, String path )
     throws SVNException {
          Collection entries = repository.getDir( path, -1 , null , (Collection) null );

          Iterator iterator = entries.iterator( );
          while ( iterator.hasNext( ) ) {
               SVNDirEntry entry = ( SVNDirEntry ) iterator.next( );
               System.out.println(
                    "/" + (path.equals( "" ) ? "" : path + "/" )
               +    entry.getName( )
               +    " ( author: '" + entry.getAuthor( )
               +    "'; revision: " + entry.getRevision( )
               +    "; date: " + entry.getDate( ) + ")" );
               if ( entry.getKind() == SVNNodeKind.DIR ) {
                    listEntries(
                         repository,
                         ( path.equals( "" ) ) ? entry.getName( ) : path
                              + "/" + entry.getName( ) );
               }
          }
}
```

4. svn list

```
SVNNodeKind nodeKind = repository.checkPath( "" ,  -1 );
if ( nodeKind == SVNNodeKind.NONE ) {
      System.err.println( "There is no entry");
      System.exit( 1 );
} else if ( nodeKind == SVNNodeKind.FILE ) {
      System.err.println( "A directory was expected." );
      System.exit( 1 );
}


listEntries( repository , "" );
```

5. svn rm/add/mv/cp Helper Class

```
private SVNRepository repository;
private ISVNEditor editor;
private long youngest;

public RepositoryOperations(SVNRepository repos, String logMsg) throws Exception {
      repository = repos;
      youngest = repository.getLatestRevision();
      editor = getCommitEditor(logMsg);
}

public void addDirs(String source, String[] dirs) throws Exception {
      editor.openRoot(-1);
      editor.openDir(source, -1);
      for (int i = 0; i < dirs.length; i++) {
            editor.addDir(source + "/" + dirs[i], null, -1);
      }
      editor.closeDir();
      editor.closeDir();.
}
```

5. svn rm/add/mv/cp Helper Class

```
public void deleteDirs(String[] dirs) throws Exception {
      editor.openRoot(-1);

      for (int i = 0; i < dirs.length; i++) {
            editor.deleteEntry(dirs[i], SVNRevision.HEAD.getNumber());
      }

      editor.closeDir();
}
```

5. svn rm/add/mv/cp Helper Class

```
public void addFiles(String source, String[] files, String[] contents)
      throws Exception {

      editor.openRoot(-1);

      editor.openDir(source, SVNRevision.HEAD.getNumber());

      for (int i = 0; i < files.length; i++) {
            editor.addFile(source + "/" + files[i], null , -1 );
            editor.applyTextDelta(source + "/" + files[i], null );
            SVNDeltaGenerator deltaGenerator = new SVNDeltaGenerator( );
            String checksum = deltaGenerator.sendDelta(source + "/" + files[i],
                  new ByteArrayInputStream(contents[i].getBytes()) , editor , true );
            editor.closeFile(source + "/" + files[i], checksum);
      }

      editor.closeDir();

      editor.closeDir();
}
```

5. svn rm/add/mv/cp Helper Class

```
public void modifyFiles(String source,
      String[] files, String[] oldContents, String[] newContents) throws Exception  {
      editor.openRoot(-1);
      editor.openDir(source, SVNRevision.HEAD.getNumber());

      for (int i = 0; i < files.length; i++) {
            editor.openFile(source + "/" + files[i], -1 );
            editor.applyTextDelta(source + "/" + files[i], null );
            SVNDeltaGenerator deltaGenerator = new SVNDeltaGenerator( );
            String checksum = deltaGenerator.sendDelta(
                  source + "/" + files[i],
                  new ByteArrayInputStream(oldContents[i].getBytes()),
                  0,
                  new ByteArrayInputStream(newContents[i].getBytes()),
                  editor,
                  true //computeChecksum
            );
            editor.closeFile(source + "/" + files[i], checksum);
      }
      editor.closeDir();
      editor.closeDir();
}
```

5. svn rm/add/mv/cp Committing changes

```
RepositoryOperations rop = new RepositoryOperations(repository, "This is a Test");
rop.addDirs("", new String[] {
      "trunk",
      "tags",
      "branches"
});
rop.Commit();


rop = new RepositoryOperations(repository, "Create the Maven structure.");
rop.addDirs("trunk",
      new String[]{
            "src",
            "src/main",
            "src/site",
            "src/test",
      }
);
rop.Commit();
```

5. svn rm/add/mv/cp Committing changes

```
rop = new RepositoryOperations(repository, "Add file content.");
rop.addFiles(
      "trunk",
      new String[] {
            "pom.xml",
            "build.xml"
      }, //The files which will be added
      new String[] {
            "This is the new POM file.",
            "This is the new build.xml file."
      } //The contents of the files
);
rop.Commit();
```

5. svn rm/add/mv/cp Committing changes

```
rop = new RepositoryOperations(repository, "Add file content.");
rop.modifyFiles(
      "trunk",
      new String[] {
            "pom.xml",
            "build.xml"
      },
      new String[] {
            "This is the new POM file.",
            "This is the new build.xml file."
      },
      new String[] {
            "This is the new POM file after we change the file",
            "This is the new build.xml file after we changed the file."
      }
);
rop.Commit();
```

On-line Sources I

- Homepage SVNKit
  - http://www.svnkit.com
- Wiki of SVNKit
  - https://wiki.svnkit.com/
- Homepage Subversion
  - http://subversion.tigris.org

Questions?

subconf2009@soebes.com

Thank you for your attention.