

# RCS

Revisions control system

# Revisions as data snapshot

```
try (CommitWriter commit = RCSManager.commit(h)) {  
    commit  
        .author("somebody")  
        .message("here go message <possible>xml</possible>")  
}
```

# Commit metadata

```
.metadata("key-string", "value")
```

```
.metadata("key-xml", "<possible>xml</possible>")
```

```
.metadata("key-json", "[{"name":"value"}]")
```

# Operations that bring data to current state

```
.create(colURL.append("test1.xml"))
```

```
.update(colURL.append("test2.xml"))
```

```
.move(colURL.append("test3.xml"))
```

```
.rename(colURL.append("test4.xml"))
```

```
.delete(colURL.append("test5.xml"))
```

# Finish - flush

```
.done();
```

```
}
```

# Reading revisions back

```
RCSResource resource = RCSManager.resource(docId);
```

```
List<Revision> revisions = resource.revisions();
```

```
Revision revision = resource .lastRevision();
```

```
Revision revision = resource.revision(id_as_long);
```

# What is it?

Class Revision {

long id();

boolean isCollection() throws IOException

boolean isXML() throws IOException

boolean isBinary() throws IOException

boolean isDeleted() throws IOException

# Get data stream

```
public InputStream getData() throws IOException
```

```
public DocumentImpl getXML(Database db) throws IOException, SAXException
```



# Read metadata

```
public <T extends MetasHandler> T metadata(T dh) throws IOException;
```

# Restore to that point

```
void restore(DBBroker broker, Handler h) throws Exception;
```

```
}
```

# Internal structure

- commits/
  - uuids/
  - hashes/
  - tmp/
  - snapshots/
- commits metadata
  - revisions
  - data storage
  - mainly used to calculate hash
  - backups

Q&A