01a: Convert XML file To Sequence File – writing & reading – Local File System

Posted on May 1, 2016

Sequence files are good for saving **raw data** into HDFS. Sequence files are **compressible** and **splittable**. It is also useful for combining a number of smaller files into a single say 64MB or larger sequence file as HDFS is more suited for larger files. Text files like csv, xml, json, etc can be stored on HDFS in **sequence file format**.

Step 1: Create a Java project using Maven.

```

mvn archetype:generate -DgroupId=com.mytutorial -D

```

Step 2: Import it into eclipse as a an "**existing maven project**".

Step 3: Add the hadoop dependencies to the **pom.xml** file.

```

<project xmlns="http://maven.apache.org/POM/4.0.0'
    xsi:schemaLocation="http://maven.apache.org/P(
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mytutorial</groupId>
    <artifactId>sequence-file</artifactId>
    <packaging>jar</packaging>
    <version>1.0-SNAPSHOT</version>
    <name>sequence-file</name>
    <url>http://maven.apache.org</url>

    <properties>
        <maven.compiler.source>1.8</maven.compile
        <maven.compiler.target>1.8</maven.compile
        <project.build.sourceEncoding>UTF-8</proj
        <junit.version>4.8.1</junit.version>
        <hadoop.version>2.7.2</hadoop.version>
    </properties>

    <dependencies>
        <!-- JUnit -->
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>${junit.version}</version>
            <scope>test</scope>
        </dependency>

        <!-- Hadoop -->
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-hdfs</artifactId>
            <version>${hadoop.version}</version>
            <exclusions>
                <exclusion>
                    <groupId>javax.servlet</group
                    <artifactId>*</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>org.apache.hadoop</groupId>
            <artifactId>hadoop-client</artifactId
            <version>${hadoop.version}</version>
            <exclusions>
                <exclusion>
                    <groupId>javax.servlet</groupI
                    <artifactId>*</artifactId>
                </exclusion>
            </exclusions>
        </dependency>

    </dependencies>

</project>

```

Step 4: Create **src/main/resources** source folder, and under that the "data" sub-folder. Create "**report.xml**", which needs to be converted to a **sequence file format**.

```

<?xml version="1.0" encoding="UTF-8"?>
<transactionReports xmlns="http://mytutorial.com/
    <transactionReport>
        <report>
            <reportNumber>9999</reportNumber>
            <createdDatetime>2015-06-15T11:29:52+
            <processedDatetime>2015-06-15T11:29:5
            <reportStatusCode>Active</reportStatu
        </report>
    </transactionReport>
</transactionReports>

```

Step 5: Write a stand-alone Java class to convert above report.xml file to report.seq. Write and then read. This example uses the "**LOCAL FILE SYSTEM**"

```

package com.mytutorial;

import java.io.File;
import java.io.IOException;
import java.net.URL;
import java.util.List;
import java.util.stream.Collectors;

import org.apache.commons.io.FileUtils;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.BytesWritable;
import org.apache.hadoop.io.IOUtils;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.SequenceFile;
import org.apache.hadoop.util.ReflectionUtils;

public class ConvertXmlToSequence {
    private static final String FILE_IN_PATH = "d
    private static final String FILE_OUT_PATH = "

    public static void main(String[] args) throws
        URL resource = ConvertXmlToSequence.class

        Configuration conf = new Configuration();

        File inputFile = new File(resource.getPat
        Path outputFile = new Path(resource.getPa

        write(conf, inputFile, outputFile); // wr
        read(conf, outputFile); // read seq file
    }

    /**
     * Write a text file to sequence file
     *
     * @param conf
     * @param inputFile
     * @param outputFile
     */
    public static void write(Configuration conf,
        SequenceFile.Writer writer = null;

        try {
            writer = SequenceFile.createWriter(co
                    SequenceFile.Writer.compressi
                    SequenceFile.Writer.keyClass(
                    SequenceFile.Writer.valueClas

            List<String> lines = FileUtils.readLi
            String xmlString = lines.stream().map(
            IntWritable key = new IntWritable(1);
            BytesWritable value = new BytesWritab
            writer.append(key, value);

        } catch (IOException e) {
            System.out.println("Error writing: "
        }

        finally {
            IOUtils.closeStream(writer);
        }
    }

    /**
     * Read a sequence file
     *
     * @param conf
     * @param sequenceFileToRead
     */
    public static void read(Configuration conf, Pa
        SequenceFile.Reader reader = null;
        try {
            reader = new SequenceFile.Reader(conf
            IntWritable keyRead = (IntWritable) Re
            BytesWritable valueRead = (BytesWritab
            while (reader.next(keyRead, valueRead)
                System.out.println("key : " + keyR
                valueRead = (BytesWritable) Reflec
            }
        } catch (IOException e) {
            System.out.println("Cannot read sequen
        }

        IOUtils.closeStream(reader);
    }
}

```

Output:

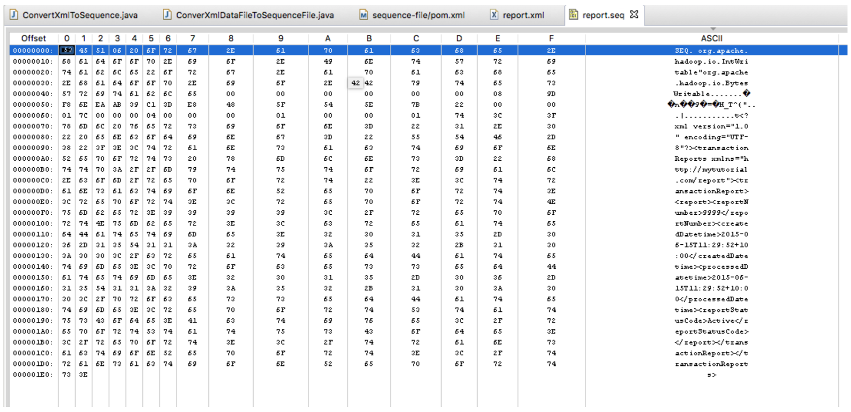
```

key : 1 - value : <?xml version="1.0" encoding="UTF

```

As you can see, the sequence files are stored as key/value pairs. The **key** being IntWritable "1" and the **value** being the "whole XML".

The **report.seq** file gets written to the project folder's "**target/classes/data**" folder. The "value" will be stored as binary.

Step 6: If you want to open the sequence file in eclipse, you need to install the hex editor plugin from

the update site:
“http://ehep.sourceforge.net/update”.



ConvertXmlToSequence.java | ConverXmlDataFileToSequenceFile.java | sequence-file/pom.xml | report.xml | report.seq

```
Offset     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   ASCII
00000000: 53 45 51 06 20 6F 72 67 2E 61 70 61 63 68 65 2E  SEQ. org.apache.
00000010: 68 61 64 6F 6F 70 2E 69 6F 2E 49 6E 74 57 72 69  hadoop.io.IntWri
00000020: 74 61 62 6C 65 22 6F 72 67 2E 61 70 61 63 68 65  table"org.apache
00000030: 2E 68 61 64 6F 6F 70 2E 69 6F 2E 42 79 74 65 73  .hadoop.io.Bytes
00000040: 57 72 69 74 61 62 6C 65 00 00 00 00 00 00 08 9D  Writable.......�
00000050: F8 6E EA AB 39 C1 3D E8 48 5F 54 5E 7B 22 00 00  �n��9�=�H_T^{"..
00000060: 01 7C 00 00 00 04 00 00 00 01 00 00 01 74 3C 3F  .|...........t<?
00000070: 78 6D 6C 20 76 65 72 73 69 6F 6E 3D 22 31 2E 30  xml version="1.0
00000080: 22 20 65 6E 63 6F 64 69 6E 67 3D 22 55 54 46 2D  " encoding="UTF-
00000090: 38 22 3F 3E 3C 74 72 61 6E 73 61 63 74 69 6F 6E  8"?><transaction
000000A0: 52 65 70 6F 72 74 73 20 78 6D 6C 6E 73 3D 22 68  Reports xmlns="h
000000B0: 74 74 70 3A 2F 2F 6D 79 74 75 74 6F 72 69 61 6C  ttp://mytutorial
000000C0: 2E 63 6F 6D 2F 72 65 70 6F 72 74 22 3E 3C 74 72  .com/report"><tr
000000D0: 61 6E 73 61 63 74 69 6F 6E 52 65 70 6F 72 74 3E  ansactionReport>
000000E0: 3C 72 65 70 6F 72 74 3E 3C 72 65 70 6F 72 74 4E  <report><reportN
000000F0: 75 6D 62 65 72 3E 39 39 39 39 3C 2F 72 65 70 6F  umber>9999</repo
00000100: 72 74 4E 75 6D 62 65 72 3E 3C 63 72 65 61 74 65  rtNumber><create
00000110: 64 44 61 74 65 74 69 6D 65 3E 32 30 31 35 2D 30  dDatetime>2015-0
00000120: 36 2D 31 35 54 31 31 3A 32 39 3A 35 32 2B 31 30  6-15T11:29:52+10
00000130: 3A 30 30 3C 2F 63 72 65 61 74 65 64 44 61 74 65  :00</createdDate
00000140: 74 69 6D 65 3E 3C 70 72 6F 63 65 73 73 65 64 44  time><processedD
00000150: 61 74 65 74 69 6D 65 3E 32 30 31 35 2D 30 36 2D  atetime>2015-06-
00000160: 31 35 54 31 31 3A 32 39 3A 35 32 2B 31 30 3A 30  15T11:29:52+10:0
00000170: 30 3C 2F 70 72 6F 63 65 73 73 65 64 44 61 74 65  0</processedDate
00000180: 74 69 6D 65 3E 3C 72 65 70 6F 72 74 53 74 61 74  time><reportStat
00000190: 75 73 43 6F 64 65 3E 41 63 74 69 76 65 3C 2F 72  usCode>Active</r
000001A0: 65 70 6F 72 74 53 74 61 74 75 73 43 6F 64 65 3E  eportStatusCode>
000001B0: 3C 2F 72 65 70 6F 72 74 3E 3C 2F 74 72 61 6E 73  </report></trans
000001C0: 61 63 74 69 6F 6E 52 65 70 6F 72 74 3E 3C 2F 74  actionReport></t
000001D0: 72 61 6E 73 61 63 74 69 6F 6E 52 65 70 6F 72 74  ransactionReport
000001E0: 73 3E                                            s>
```

Sequence File opened with a hex editor

if you write to HDFS, you can display it on HDFS shell with:

```

$ hdfs dfs -text path/report.seq

```

‹ 02: Q7 – Q15 Hadoop overview & architecture interview Q&As

02: Convert XML file To Sequence File with Apache Spark – writing & reading ›

Disclaimer