

16.7. Java program to extract records from the NPEM metadata database and write one XML file for each record.

[A copy of this file (FGDCEXtract.java) can be found at the PICES web site.]

```
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.net.InetAddress;
import java.sql.*;
import java.util.*;

import gov.noaa.foci.mdb.MDBSearchCommandHelper;

/**
 * Extracts FGDC data elements from North Pacific Metadata database
 * Creation date: (08/21/2004)
 * Modified date: (11/07/2004)
 * @author: dank
 */
public class FGDCEXtract {

    // local member variables
    private Vector title = new Vector();
    private Vector abstract = new Vector();
    private Vector purpose = new Vector();
    private Vector originator = new Vector();
    private Vector pubdate = new Vector();
    private Vector tpcdate = new Vector();
    private Vector tpcmdt = new Vector();
    private Vector tpcrdtbegdate = new Vector();
    private Vector tpcrdtenddate = new Vector();
    private Vector currref = new Vector();
    private Vector statprog = new Vector();
    private Vector statmuf = new Vector();
    private Vector sdnbc = new Vector();
    private Vector sdwbc = new Vector();
    private Vector sdebc = new Vector();
    private Vector sdsbc = new Vector();
    private Vector accessconstraint = new Vector();
    private Vector useconstraint = new Vector();
    private Vector sname = new Vector();
    private Vector sat = new Vector();
    private Vector sadd = new Vector();
    private Vector scity = new Vector();
    private Vector ssop = new Vector();
    private Vector szip = new Vector();
    private Vector scountry = new Vector();
    private Vector sphone = new Vector();
    private Vector sfax = new Vector();
    private Vector sem = new Vector();
    private Vector fgdcdate = new Vector();
    private Vector entrypersionname = new Vector();
    private Vector mdbid = new Vector();

    private Vector keyword = new Vector();
    private Vector thesaurus = new Vector();
```

```

private String dataDir = "";

// total records in the BSEMetaDatabase table
private int tablesize = 0;

// SQL members
private ResultSet result;
private Connection conn = null;

// Helper class
private MDBSearchCommandHelper helper = new MDBSearchCommandHelper();

// Query strings
private String query1 = "SELECT Title, Abstract, Purpose, " +
"Originator, PubDate, TPCDate, TPCMDT, TPCRDTBegDate, " +
"TPCRDTEndDate, CurRef, StatProg, StatMUF, SDNBC, SDWBC, SDEBC, SDSBC,
" +
"AccConstraint, UseConstraint, SName, SAT, SAdd, SCity, SSOP, " +
"SZIP, SCountry, SPhone, SFax, SEM, FGDCDate, EntryPersonName, MdbId FROM
bsemetadata";

private String query2 = "SELECT Thesaurus, Keyword FROM keywords WHERE MdbID
= ";

private String query3 = "";

private String distLiab = " While every effort has been made to ensure that
these data " +
"are accurate and reliable within the limits of the current state of the art,
NOAA cannot " +
"assume liability for any damages caused by any errors or omissions in the
data, nor as a " +
"result of the failure of the data to function on a particular system. NOAA
makes no warranty, " +
"expressed or implied, nor does the fact of distribution constitute such a
warranty.";

/**
 * execute
 * This is the work horse method for the command.
 * It will execute the queries and get the result set.
 */
public void execute()
throws
java.lang.Exception,
java.io.IOException,
java.sql.SQLException {

    try {
        Counter counter = new Counter();
        tablesize = counter.countTotalRecords(conn);
        Searcher searcher = new Searcher();
        searcher.searchBSEMetaDataTable(conn, query1);
        for (int i=0; i<tablesize; i++) {
            thesaurus.clear();
            keyword.clear();
query3 = query2 + (String) mdbid.elementAt(i) + " ORDER BY Thesaurus ASC";

```

```

        searcher.searchKeywordTable(conn, query3);
        writeFGDCXML(i);
    }
} catch (Throwable theException) {
    theException.printStackTrace();
} finally {
    conn.close();
}
}

public class Counter {
private static final String COUNT_SQL = "SELECT Count(*) AS trecs FROM
bsemetadata";
    public int countTotalRecords(Connection con) {
        int size = 0;
        try {
            Statement statement = conn.createStatement();
            ResultSet rs = statement.executeQuery(COUNT_SQL);
            while (rs.next()) {
                size = rs.getInt("trecs");
            }
        } catch (Throwable theException) {
            theException.printStackTrace();
        }
        return size;
    }
}

public class Searcher {
    public void searchBSEMetaDataTable(Connection con, String query) {
        try {
            Statement statement = conn.createStatement();
            ResultSet rs = statement.executeQuery(query);
            while (rs.next()) {
                title.addElement(rs.getString(1));
                abstrct.addElement(rs.getString(2));
                purpose.addElement(rs.getString(3));
                originator.addElement(rs.getString(4));
                pubdate.addElement(rs.getString(5));
                tpcdate.addElement(rs.getString(6));
                tpcmdt.addElement(rs.getString(7));
                tpcrdtbegdate.addElement(rs.getString(8));
                tpcrdtenddate.addElement(rs.getString(9));
                currref.addElement(rs.getString(10));
                statprog.addElement(rs.getString(11));
                statmuf.addElement(rs.getString(12));
                sdnbc.addElement(rs.getString(13));
                sdwbc.addElement(rs.getString(14));
                sdebc.addElement(rs.getString(15));
                sdsbc.addElement(rs.getString(16));

accessconstraint.addElement(rs.getString(17));
                useconstraint.addElement(rs.getString(18));
                sname.addElement(rs.getString(19));
                sat.addElement(rs.getString(20));
                sadd.addElement(rs.getString(21));
                scity.addElement(rs.getString(22));
            }
        }
    }
}

```

```

        ssop.addElement(rs.getString(23));
        szip.addElement(rs.getString(24));
        scountry.addElement(rs.getString(25));
        sphone.addElement(rs.getString(26));
        sfax.addElement(rs.getString(27));
        sem.addElement(rs.getString(28));
        fgdcdate.addElement(rs.getString(29));
        entriypersonname.addElement(rs.getString(30));
        mdbid.addElement(rs.getString(31));
    }
} catch (Throwable theException) {
    theException.printStackTrace();
}
}

public void searchKeywordTable(Connection con, String query) {
try {
    Statement statement = conn.createStatement();
    ResultSet rs = statement.executeQuery(query);
    while (rs.next()) {
        thesaurus.addElement(rs.getString(1));
        keyword.addElement(rs.getString(2));
    }
} catch (Throwable theException) {
    theException.printStackTrace();
}
}

private String dataDir() {
String dir = "";
try {
    InetAddress localaddr = InetAddress.getLocalHost();
if (localaddr.getHostName().equals("afscmaps.akctr.noaa.gov") ||
localaddr.getHostName().equals("afscmaps.afsc.noaa.gov")) { // production
    dir="/usr/local/Isite2/data/";
    } else {
if (localaddr.getHostName().equals("gandalf")) { // Dan's RH9.0 box
    dir = "/usr/local/Isite2/data/";
    } else {
        if (localaddr.getHostName().equals("pippin")) { //
Dan's Win2K box
            dir = "c:\\fgdc\\";
        } else {
System.out.println("Could not set the directory from the hostname");
        }
    }
} catch (Throwable t) {
    t.printStackTrace();
}
return dir;
}

private void writeFGDCXML(int index) {
    dataDir = dataDir();
    try {

```

```

FileOutputStream fout = new FileOutputStream(dataDir + "npem" + index +
".xml");
    OutputStreamWriter out = new OutputStreamWriter(fout, "UTF-8");
    out.write("<?xml version=\"1.0\"?>\r\n");
    out.write("<metadata>\r\n");
    out.write("    <idinfo>\r\n");
    out.write("        <citation>\r\n");
    out.write("            <citeinfo>\r\n");
        out.write("                <origin>" + originator.elementAt(index) +
"</origin>\r\n");
        out.write("                    <pubdate>" + pubdate.elementAt(index) +
"</pubdate>\r\n");
        out.write("                        <title>" + title.elementAt(index) +
"</title>\r\n");
        out.write("                            </citeinfo>\r\n");
        out.write("                                </citation>\r\n");
        out.write("                                    <descript>\r\n");
        out.write("                                        <abstract>" + abstrct.elementAt(index) +
"</abstract>\r\n");
        out.write("                                            <purpose>" + purpose.elementAt(index) +
"</purpose>\r\n");
        out.write("                                                </descript>\r\n");
        out.write("                                                    <timeperd>\r\n");
        out.write("                                                        <timeinfo>\r\n");
if (!(tpcdate.elementAt(index) == null ||
tpcdate.elementAt(index).equals("")))) {
        out.write("                    <sngdate>\r\n");
        out.write("                        <caldate>" + tpcdate.elementAt(index) +
"</caldate>\r\n");
        out.write("                            </sngdate>\r\n");
} else if (!(tpcmdt.elementAt(index) == null ||
tpcmdt.elementAt(index).equals("")))) {
        out.write("                <mdattim>\r\n");
        out.write("                    <sngdate>\r\n");
        out.write("                        <caldate>" + tpcmdt.elementAt(index) +
"</caldate>\r\n");
        out.write("                            </sngdate>\r\n");
        out.write("                                </mdattim>\r\n");
        } else if (!(tpcrdtbegdate.elementAt(index) == null ||
tpcrdtbegdate.elementAt(index).equals("")))) {
        out.write("                <rngdates>\r\n");
        out.write("                    <begdate>" + tpcrdtbegdate.elementAt(index)
+
"</begdate>\r\n");
        out.write("                        <enddate>" + tpcrdtenddate.elementAt(index)
+
"</enddate>\r\n");
        out.write("                            </rngdates>\r\n");
        }
        out.write("            </timeinfo>\r\n");
        out.write("                <current>" + currref.elementAt(index) +
"</current>\r\n");
        out.write("                    </timeperd>\r\n");
        out.write("                        <status>\r\n");
        out.write("                            <progress>" + statprog.elementAt(index) +
"</progress>\r\n");

```

```

        out.write("        <update>" + statmuf.elementAt(index) +
"</update>\r\n");
        out.write("        </status>\r\n");
        out.write("        <spdom>\r\n");
        out.write("        <bounding>\r\n");
        out.write("        <westbc>" + sdwbc.elementAt(index) +
"</westbc>\r\n");
        out.write("        <eastbc>" + sdebc.elementAt(index) +
"</eastbc>\r\n");
        out.write("        <northbc>" + sdnbc.elementAt(index) +
"</northbc>\r\n");
        out.write("        <southbc>" + sdsbc.elementAt(index) +
"</southbc>\r\n");
        out.write("        </bounding>\r\n");
        out.write("        </spdom>\r\n");
        if (!thesaurus.isEmpty()) {
            out.write("        <keywords>\r\n");
            String lastThesaurus = "";
            for (int i = 0; i < thesaurus.size(); i++) {
                if (!((String)thesaurus.elementAt(i)).equals(lastThesaurus)) {
                    if (!lastThesaurus.equals("")) out.write("</theme>\r\n");
                    out.write("        <theme>\r\n");
                    out.write("        <themekt>" +
thesaurus.elementAt(i) + "</themekt>\r\n");
                }
                out.write("        <themekey>" + keyword.elementAt(i) +
"</themekey>\r\n");
                lastThesaurus = (String) thesaurus.elementAt(i);
            }
            out.write("        </theme>\r\n");
            out.write("        </keywords>\r\n");
        }
        out.write("        <accconst>" + accessconstraint.elementAt(index) +
"</accconst>\r\n");
        out.write("        <useconst>" + useconstraint.elementAt(index) +
"</useconst>\r\n");
        out.write("        <ptcontac>\r\n");
        out.write("        <cntinfo>\r\n");
        out.write("        <cntperp>\r\n");
        out.write("        <cntper>" + sname.elementAt(index) +
"</cntper>\r\n");
        out.write("        </cntperp>\r\n");
        out.write("        <cntaddr>\r\n");
        out.write("        <addrtype>" + sat.elementAt(index) +
"</addrtype>\r\n");
        out.write("        <address>" + sadd.elementAt(index) +
"</address>\r\n");
        out.write("        <city>" + scity.elementAt(index) +
"</city>\r\n");
        out.write("        <state>" + ssop.elementAt(index) +
"</state>\r\n");
        out.write("        <postal>" + szip.elementAt(index) +
"</postal>\r\n");
        out.write("        <country>" + scountry.elementAt(index) +
"</country>\r\n");
        out.write("        </cntaddr>\r\n");
        out.write("        <cntvoice>" + sphone.elementAt(index) +

```

```

" </cntvoice>\r\n");
    out.write("          <cntfax>" + sfax.elementAt(index) +
" </cntfax>\r\n");
    out.write("          <cntemail>" + sem.elementAt(index) +
" </cntemail>\r\n");
    out.write("          </cntinfo>\r\n");
    out.write("          </ptcontac>\r\n");
    out.write("        </idinfo>\r\n");
    out.write("        <distinfo>\r\n");
    out.write("        <distrib>\r\n");
    out.write("        <cntinfo>\r\n");
    out.write("        <cntperp>\r\n");
    out.write("        <cntper>" + sname.elementAt(index) +
" </cntper>\r\n");
    out.write("        </cntperp>\r\n");
    out.write("        <cntaddr>\r\n");
    out.write("        <addrtype>" + sat.elementAt(index) +
" </addrtype>\r\n");
    out.write("        <address>" + sadd.elementAt(index) +
" </address>\r\n");
    out.write("        <city>" + scity.elementAt(index) +
" </city>\r\n");
    out.write("        <state>" + ssop.elementAt(index) +
" </state>\r\n");
    out.write("        <postal>" + szip.elementAt(index) +
" </postal>\r\n");
    out.write("        <country>" + scountry.elementAt(index) +
" </country>\r\n");
    out.write("        </cntaddr>\r\n");
    out.write("        <cntvoice>" + sphone.elementAt(index) +
" </cntvoice>\r\n");
    out.write("        <cntfax>" + sfax.elementAt(index) +
" </cntfax>\r\n");
    out.write("        <cntemail>" + sem.elementAt(index) +
" </cntemail>\r\n");
    out.write("        </cntinfo>\r\n");
    out.write("        </distrib>\r\n");
    out.write("        <distliab>" + distLiab + "</distliab>\r\n");
    out.write("        </distinfo>\r\n");
    out.write("        <metainfo>\r\n");
    out.write("        <metd>" + fgdcdate.elementAt(index) + "</metd>\r\n");
    out.write("        <metc>\r\n");
    out.write("        <cntinfo>\r\n");
    out.write("        <cntperp>\r\n");
    out.write("        <cntper>Kimberly Bahl</cntper>\r\n");
    out.write("        </cntperp>\r\n");
    out.write("        <cntaddr>\r\n");
    out.write("        <addrtype>Mailing and Physical
Address</addrtype>\r\n");
    out.write("        <address> NOAA / AFSC / PMEL, 7600 Sand Point
Way
N.E.</address>\r\n");
    out.write("        <city>Seattle</city>\r\n");
    out.write("        <state>WA</state>\r\n");
    out.write("        <postal>98115</postal>\r\n");
    out.write("        <country>USA</country>\r\n");
    out.write("        </cntaddr>\r\n");

```

```

        out.write("        <cntvoice>206-526-4314</cntvoice>\r\n");
        out.write("        <cntfax>206-526-6485</cntfax>\r\n");
        out.write("
<cntemail>kimberly.bahl@noaa.gov</cntemail>\r\n");
        out.write("        </cntinfo>\r\n");
        out.write("        </metc>\r\n");
        out.write("        <metstdn>Content Standards for Digital Geospatial
Metadata</metstdn>\r\n");
        out.write("        <metstdv>FGDC-STD-001-1998</metstdv>\r\n");
        out.write("        <metac>" + accessconstraint.elementAt(index) +
"</metac>\r\n");
        out.write("        <metuc>" + useconstraint.elementAt(index) +
"</metuc>\r\n");
        out.write("    </metainfo>\r\n");
        out.write("</metadata>\r\n");
        out.close();
    } catch (IOException e) {
        System.err.println(e);
    }
}

/**
 * initialize
 * This method will connect to the database.
 */
public void initialize() throws java.io.IOException {
    conn = helper.connectToDB();
}

public int getTableSize() {
    return tablesize;
}

public String getDataDir() {
    return dataDir;
}
}

```