Countries Of The World App 1.5 - Java

CS 3310 – Dr. Donna Kaminski

Index

[Main 2](#_Toc379858577)

[Setup 3](#_Toc379858578)

[UserApp 4](#_Toc379858579)

[RawData 6](#_Toc379858580)

[TransData 8](#_Toc379858581)

[CountryDataTable 11](#_Toc379858582)

[CountryDataRec 15](#_Toc379858583)

[TheLog 19](#_Toc379858584)

[TheLog FILE 22](#_Toc379858585)

**package** edu.wmich.cs3310.asgn1;

**import** java.io.File;

**import** java.io.IOException;

/\*\*

\* Countries Of The World App 1.0

\* Creates table for managing contries information.

\* **@author** Caleb Viola

\*/

# **public** **class** Main {

/\*\*

\* Deletes binary file if exists, calls Setup and Userapp according to

\* A2DemoSpecs.

\* **@param** args

\* **@throws** IOException

\*/

**public** **static** **void** main(String[] args) **throws** IOException{

File file = **new** File("CountryData.bin");

**if** (file.exists())

file.delete();

Setup.*Table*("A2");

**for** (**int** i = 5; i <= 7; i++)

UserApp.*Table*(i);

}

}

/\*\*

\*

\*/

**package** edu.wmich.cs3310.asgn1;

**import** java.io.IOException;

/\*\*

\* Class containing method to setup CDT from the RawData files.

\* **@author** Caleb Viola

\*/

# **public** **class** Setup {

/\*\*

\* Method intented to setup CDT from the RawData files.

\* **@throws** IOException

\*/

**public** **static** **void** Table(String fileNameSufix) **throws** IOException {

TheLog tl = **new** TheLog(**false**);

tl.displayThis("CODE STATUS > Setup started.");

CountryDataTable cdt = **new** CountryDataTable(tl, "CountryData.bin");

RawData rd = **new** RawData(fileNameSufix, tl);

**int** temp = 0;

**while** (!rd.doneWithInput) {

rd.GrabCountry();

**if** (!rd.doneWithInput)

cdt.Insert1Country(rd.GetID(), rd.GetCode(), rd.GetName(),

rd.GetContinent(), rd.GetArea(), rd.GetPopulation(),

rd.GetLifeExpectancy(), tl, **false**);

}

rd.FinishUp(tl);

cdt.FinishUp(tl, **true**);

tl.displayThis("CODE STATUS > Setup finished - " + temp

+ " countries processed.");

tl.FinishUp();

}

}

**package** edu.wmich.cs3310.asgn1;

**import** java.io.IOException;

/\*\*

\* Class containing method to manage table modifinations from the TransData files.

\* **@author** Caleb Viola

\*/

# **public** **class** UserApp {

/\*\*

\* Method intented to manage table modifinations from the TransData files.

\* **@param** fileNameSufix

\* **@throws** IOException

\*/

**public** **static** **void** Table(**int** fileNameSufix)**throws** IOException {

TheLog tl = **new** TheLog(**true**);

tl.displayThis("CODE STATUS > UserApp started.");

CountryDataTable cdt = **new** CountryDataTable(tl, "CountryData.bin");

TransData td = **new** TransData(fileNameSufix, tl);

**int** temp = 0;

**while** (!td.doneWithFile) {

td.GrabCommand(tl);

**if** (!td.doneWithFile) {

**switch** (td.TransCode()) {

**case** "SI":

**if** (td.GetID() == -1) {

tl.displayThis("SI ERROR");

tl.displayThis(tl.Sorry3());

temp++;

**break**;

}

tl.displayThis("SI " + td.GetID());

cdt.SelectByID(td.GetID(), tl);

temp++;

**break**;

**case** "SA":

tl.displayThis("SA");

cdt.SelectAll(tl);

temp++;

**break**;

**case** "IN":

tl.displayThis("IN " + td.GetID() + "," + td.GetCode()

+ "," + td.GetName() + "," + td.GetArea() +

"," + td.GetPopulation() + "," + td.GetLifeExpectancy());

cdt.Insert1Country(td.GetID(), td.GetCode(), td.GetName(),

td.GetContinent(), td.GetArea(),

td.GetPopulation(), td.GetLifeExpectancy()

tl, **true**);

temp++;

**break**;

**case** "DI":

**if** (td.GetID() == -1) {

tl.displayThis("DI ERROR");

tl.displayThis(tl.Sorry3());

temp++;

**break**;

}

tl.displayThis("DI " + td.GetID());

cdt.DeleteByID(td.GetID(), tl);

temp++;

**break**;

**default**:

tl.displayThis(td.TransCode() + "\n"

+ " ERROR, invalid command.");

temp++;

}

}

}

td.FinishUp(tl);

cdt.FinishUp(tl, **true**);

tl.displayThis("CODE STATUS > UserApp finished - " + temp

+ " transactions processed.");

tl.FinishUp();

}

}

**package** edu.wmich.cs3310.asgn1;

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.Scanner;

/\*\*

\* Intended to obtain information from the RawData files for the Setup class.

\* Countries Of The World App 1.0

\* **@author** Caleb Viola

\*/

# **public** **class** RawData {

**private** Scanner input;

**private** **short** id;

**private** String code;

**private** String name;

**private** String continent;

**private** **int** area;

**private** **long** population;

**private** **float** lifeExpectancy;

**boolean** doneWithInput = **false**;

/\*\*

\* Contructor to open RawData and create log entry.

\* **@param** fileNameSufix Name of RawData file to open.

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**public** RawData(String fileNameSufix, TheLog tl) **throws** IOException {

File file = **new** File("RawData" + fileNameSufix + ".csv");

tl.displayThis("FILE STATUS > RawData FILE opened.");

input = **new** Scanner(file);

}

/\*\*

\* Obtain information from one line in the RawData file.

\*/

**public** **void** GrabCountry() {

**if** (input.hasNextLine()) {

String temp = input.nextLine();

temp = temp.substring(30, temp.length() - 2).replace("'", "");

String[] fields = temp.split(",");

id = Short.*parseShort*(fields[0]);

code = fields[1];

name = fields[2];

continent = fields[3];

area = Integer.*parseInt*(fields[5]);

population = Integer.*parseInt*(fields[7]);

lifeExpectancy = Float.*parseFloat*(fields[8]);

} **else**

doneWithInput = **true**;

}

/\*\*

\* Getter for id.

\* **@return**

\*/

**public** **short** GetID(){

**return** id;

}

/\*\*

\* Getter for code.

\* **@return**

\*/

**public** String GetCode(){

**return** code;

}

/\*\*

\* Getter for name.

\* **@return**

\*/

**public** String GetName(){

**return** name;

}

/\*\*

\* Getter for continent.

\* **@return**

\*/

**public** String GetContinent(){

**return** continent;

}

/\*\*

\* Getter for area.

\* **@return**

\*/

**public** **int** GetArea(){

**return** area;

}

/\*\*

\* Getter for population.

\* **@return**

\*/

**public** **long** GetPopulation(){

**return** population;

}

/\*\*

\* Getter for lifeExpectancy.

\* **@return**

\*/

**public** **float** GetLifeExpectancy(){

**return** lifeExpectancy;

}

/\*\*

\* Close file and create log entry.

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**public** **void** FinishUp(TheLog tl) **throws** IOException{

input.close();

tl.displayThis("FILE STATUS > RawData FILE closed.");

}

}

**package** edu.wmich.cs3310.asgn1;

**import** java.io.File;

**import** java.io.IOException;

**import** java.util.Scanner;

/\*\*

\* Intended to obtain information for UserApp in Main from the TransData files

\* in order to create table.

\* **@author** Caleb Viola

\*/

# **public** **class** TransData {

**private** Scanner input;

**private** **short** id;

**private** String code;

**private** String name;

**private** String continent;

**private** String command;

**private** **int** area;

**private** **long** population;

**private** **float** lifeExpectancy;

**boolean** doneWithFile = **false**;

/\*\*

\* Constructor to open a TransData file and create log entry.

\* **@param** fileNameSufix number of TransData file to read.

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**public** TransData(**int** fileNameSufix, TheLog tl) **throws** IOException {

File file = **new** File("TransData" + fileNameSufix + ".txt");

tl.displayThis("FILE STATUS > TransData FILE opened.");

input = **new** Scanner(file);

}

/\*\*

\* Obtain command from a line in the TransData file.

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**public** **void** GrabCommand(TheLog tl) **throws** IOException {

**if** (input.hasNextLine()) {

String temp = input.nextLine();

command = temp.substring(0, 2);

**if** (command.equals("IN")) {

temp = temp.substring(33, temp.length() - 2)

.replace("'", "");

String[] fields = temp.split(",");

id = Short.*parseShort*(fields[0]);

code = fields[1];

name = fields[2];

continent = fields[3];

area = Integer.*parseInt*(fields[5]);

population = Integer.*parseInt*(fields[7]);

lifeExpectancy = Float.*parseFloat*(fields[8]);

} **else** **if** (command.equals("SI") || command.equals("DI")) {

**if** (temp.length() > 2) {

**try** {

id = Short.*parseShort*(

temp.substring(3, temp.length()).trim());

} **catch** (NumberFormatException e) {

id = -1;

}

} **else**

id = -1;

}

} **else**

doneWithFile = **true**;

}

/\*\*

\* Getter for acquired command.

\* **@return**

\*/

**public** String TransCode() {

**return** command;

}

/\*\*

\* Getter for id.

\* **@return**

\*/

**public** **short** GetID() {

**return** id;

}

/\*\*

\* Getter for code.

\* **@return**

\*/

**public** String GetCode() {

**return** code;

}

/\*\*

\* Getter for name.

\* **@return**

\*/

**public** String GetName() {

**return** name;

}

/\*\*

\* Getter for continent.

\* **@return**

\*/

**public** String GetContinent() {

**return** continent;

}

/\*\*

\* Getter for area.

\* **@return**

\*/

**public** **int** GetArea() {

**return** area;

}

/\*\*

\* Getter for population.

\* **@return**

\*/

**public** **long** GetPopulation() {

**return** population;

}

/\*\*

\* Getter for lifeExpectancy.

\* **@return**

\*/

**public** **float** GetLifeExpectancy() {

**return** lifeExpectancy;

}

/\*\*

\* Close file and creat log entry.

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**public** **void** FinishUp(TheLog tl) **throws** IOException {

input.close();

tl.displayThis("FILE STATUS > TransData FILE closed.");

}

}

**package** edu.wmich.cs3310.asgn1;

**import** java.io.IOException;

**import** java.io.RandomAccessFile;

**import** java.text.DecimalFormat;

/\*\*

\* Manages countries by implementing a table

\* stored in a binary file.

\* Countries Of The World App 1.0

\* **@author** Caleb Viola

\*/

# **public** **class** CountryDataTable {

**private** **short** N;

**private** **boolean**[] status;

**private** RandomAccessFile file;

**private** CountryDataRec cdr;

/\*\*

\* Creates binaryfile and creates CountryDataRec object.

\* **@param** tl

\* **@param** fileName

\* **@throws** IOException

\*/

**public** CountryDataTable(TheLog tl, String fileName) **throws** IOException{

file = **new** RandomAccessFile(fileName, "rw");

file.seek(0);

**try** {

N = file.readShort();

} **catch** (IOException e) {

N = 0;

}

tl.displayThis("FILE STATUS > CountryData FILE opened.");

cdr = **new** CountryDataRec();

}

/\*\*

\* Execution for SI command which locates an

\* element in a binary file using id.

\* **@param** Element id to locate

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**public** **void** SelectByID(**int** id, TheLog tl) **throws** IOException{

status = cdr.byteOffsetAndStatus(id, file);

**if**(id == 0)

tl.displayThis(tl.Sorry());

**else** **if** (status[0] && !status[1]){

cdr.Read1Country(file);

tl.displayThis(" "

+ tl.Country(cdr.getCode(), cdr.getName(),

cdr.getContinent(), cdr.getArea(),

cdr.getPopulation(), cdr.getLifeExpectancy()));

}

**else** **if** (!status[0] && !status[1])

tl.displayThis(tl.Sorry3());

**else**

tl.displayThis(tl.Sorry());

}

/\*\*

\* Execution for SA command which displays table.

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**public** **void** SelectAll(TheLog tl) **throws** IOException{

tl.displayThis(" "+tl.Header1());

**short** i = 1;

status = cdr.byteOffsetAndStatus(i, file);

**while** (!(status[0] && status[1])){

cdr.Read1Country(file);

**if**(cdr.getID() != 0)

tl.displayThis("

"+tl.Country(cdr.getCode(),cdr.getName(),

cdr.getContinent(), cdr.getArea(), cdr.getPopulation(),

cdr.getLifeExpectancy()));

status = cdr.byteOffsetAndStatus(i++, file);

}

tl.displayThis(" "+tl.Footer());

}

/\*\*

\* Execution for IN command which inserts country in table.

\* **@param** id

\* **@param** code

\* **@param** name

\* **@param** continent

\* **@param** area

\* **@param** population

\* **@param** lifeExpectancy

\* **@param** tl TheLog object

\* **@param** userApp boolean to determine if confirmation message should be logged

\* **@throws** IOException

\*/

**public** **void** Insert1Country(**short** id, String code, String name,

String continent, **int** area, **long** population, **float** lifeExpectancy, TheLog tl, **boolean** userApp) **throws** IOException {

status = cdr.byteOffsetAndStatus(id, file);

**if** ((status[0] && status[1]) || (!status[0] && status[1])){

cdr.Write1Country(file, id, code, name, continent, area,

population, lifeExpectancy);

N++;

**if** (userApp)

tl.displayThis(tl.Msg("inserted"));

}

**else** **if** (status[0] && !status[1])

tl.displayThis(tl.Sorry2());

**else**

tl.displayThis(tl.Sorry3());

}

/\*\*

\* Deletes country in table by changing it's id to 0.

\* **@param** id

\* **@param** tl

\* **@throws** IOException

\*/

**public** **void** DeleteByID(**int** id, TheLog tl) **throws** IOException{

status = cdr.byteOffsetAndStatus(id, file);

**if**(id == 0)

tl.displayThis(tl.Sorry());

**else** **if** (status[0] && !status[1]){

file.seek(cdr.getByteOffset());

file.writeShort(0);

tl.displayThis(tl.Msg("deleted"));

N--;

}

**else** **if** (!status[0] && !status[1])

tl.displayThis(tl.Sorry3());

**else**

tl.displayThis(tl.Sorry());

}

/\*\*

\* Prints all contries and empty locations.

\* **@param** tl TheLog object

\* **@throws** IOException

\*/

**private** **void** Snapshot(TheLog tl) **throws** IOException{

tl.displayThis("CODE STATUS > Snapshot started.");

DecimalFormat noFormat = **new** DecimalFormat("#000");

tl.displayThis("N: "+N);

tl.displayThis(tl.Header2());

**short** i = 1;

status = cdr.byteOffsetAndStatus(i, file);

**while** (!(status[0] && status[1])){

cdr.Read1Country(file);

**if**(cdr.getID() != 0)

tl.displayThis(tl.Country2(i,cdr.getCode(),cdr.getName(), cdr.getContinent(), cdr.getArea(), cdr.getPopulation(), cdr.getLifeExpectancy()));

**else**

tl.displayThis("["+noFormat.format(i)+"] "+"

EMPTY");

i++;

status = cdr.byteOffsetAndStatus(i, file);

}

tl.displayThis(tl.Footer()+tl.Footer2());

tl.displayThis("CODE STATUS > Snapshot finished - "+N+" countries

displayed.");

}

/\*\*

\* Records N in table, calls Snapshot, closes binary file.

\* **@param** tl TheLog object

\* **@param** printTable

\* **@throws** IOException

\*/

**public** **void** FinishUp(TheLog tl, **boolean** printTable) **throws** IOException{

file.seek(0);

file.writeShort(N);

**if** (printTable)

Snapshot(tl);

file.close();

tl.displayThis("FILE STATUS > CountryData FILE closed.");

}

}

**package** edu.wmich.cs3310.asgn1;

**import** java.io.IOException;

**import** java.io.RandomAccessFile;

/\*\*

\* Intended to manage binary file handling.

\* **@author** Caleb Viola

\*/

# **public** **class** CountryDataRec {

**private** **short** id;

**private** String code ;

**private** String name;

**private** String continent;

**private** **int** area;

**private** **long** population;

**private** **float** lifeExpectancy;

**private** **int** byteOffset;

**private** **int** sizeOfHeaderRec = 2;

**private** **int** sizeOfDataRec = 2 + (2 \* 3) + (2 \* 16) + (2 \* 13) + 4 + 8 + 4;

/\*\*

\* For reading a country from record.

\* **@param** file

\* **@throws** IOException

\*/

**public** **void** Read1Country(RandomAccessFile file) **throws** IOException{

code = "";

name = "";

continent = "";

id = file.readShort();

**for** (**int** i = 0; i < 3; i++)

code += file.readChar();

**for** (**int** i = 0; i < 16; i++)

name += file.readChar();

**for** (**int** i = 0; i < 13; i++)

continent += file.readChar();

area = file.readInt();

population = file.readLong();

lifeExpectancy = file.readFloat();

}

/\*\*

\* For writing one country to record.

\* **@param** file

\* **@param** id

\* **@param** code

\* **@param** name

\* **@param** continent

\* **@param** area

\* **@param** population

\* **@param** lifeExpectancy

\* **@throws** IOException

\*/

**public** **void** Write1Country(RandomAccessFile file, **short** id,

String code, String name, String continent, **int** area,

**long** population, **float** lifeExpectancy) **throws** IOException {

file.writeShort(id);

**if** (code.length() < 3)

**while** (code.length() != 3)

code += " ";

**else** **if** (code.length() > 3)

code = code.substring(0, 3);

file.writeChars(code);

**if** (name.length() < 16)

**while** (name.length() != 16)

name += " ";

**else** **if** (name.length() > 16)

name = name.substring(0, 16);

file.writeChars(name);

**if** (continent.length() < 13)

**while** (continent.length() != 13)

continent += " ";

**else** **if** (continent.length() > 13)

continent = continent.substring(0, 13);

file.writeChars(continent);

file.writeInt(area);

file.writeLong(population);

file.writeFloat(lifeExpectancy);

}

/\*\*

\* For calculating and locating byteOffset, and returning the status

\* of file at given address utilizing a booleans.

\* **@param** RRN

\* **@param** file

\* **@return** spot status.

\* **@throws** IOException

\*/

**public** **boolean**[] byteOffsetAndStatus(**int** RRN, RandomAccessFile file){

byteOffset = sizeOfHeaderRec + ((RRN - 1) \* sizeOfDataRec);

**try** {

file.seek(byteOffset);

**try** {

**int** test = file.readShort();

file.seek(byteOffset);

**if** (test > 0)

**return** **new** **boolean**[] { **true**, **false** }; // Existent id.

**return** **new** **boolean**[] { **false**, **true** }; // Empty space exists

// within existent spots. } **catch** (IOException e) {

**return** **new** **boolean**[] { **true**, **true** };// New spot; not

between others.

}

} **catch** (IOException e) {

**return** **new** **boolean**[] { **false**, **false** }; // Out of range.

}

}

/\*\*

\* Getter for id.

\* **@return**

\*/

**public** **short** getID(){

**return** id;

}

/\*\*

\* Getter for code.

\* **@return**

\*/

**public** String getCode(){

**return** code;

}

/\*\*

\* Getter for name.

\* **@return**

\*/

**public** String getName(){

**return** name;

}

/\*\*

\* Getter for continent.

\* **@return**

\*/

**public** String getContinent(){

**return** continent;

}

/\*\*

\* Getter for area.

\* **@return**

\*/

**public** **int** getArea(){

**return** area;

}

/\*\*

\* Getter for population.

\* **@return**

\*/

**public** **long** getPopulation(){

**return** population;

}

/\*\*

\* Getter for lifeExpectancy.

\* **@return**

\*/

**public** **float** getLifeExpectancy(){

**return** lifeExpectancy;

}

/\*\*

\* Getter for byteOffset.

\* **@return**

\*/

**public** **int** getByteOffset() {

**return** byteOffset;

}

}

**package** edu.wmich.cs3310.asgn1;

**import** java.io.\*;

**import** java.text.DecimalFormat;

/\*\*

\* Writes log entries to a .txt file

\* Countries Of The World App 1.0

\* **@author** Caleb Viola

\*/

# **public** **class** TheLog {

**private** PrintWriter theLog;

/\*\*

\* Intended to set up theLog according to boolean for append.

\* **@param** append Decides append mode for PrintWriter.

\* **@throws** IOException

\*/

**public** TheLog(**boolean** append) **throws** IOException {

**if** (!append)

theLog = **new** PrintWriter("TheLog.txt");

**else**

theLog = **new** PrintWriter(**new** FileOutputStream(

**new** File("TheLog.txt"), append));

theLog.println("FILE STATUS > TheLog FILE opened.");

}

/\*\*

\* Preformatted string

\* **@param** code

\* **@param** name

\* **@param** continent

\* **@param** area

\* **@param** population

\* **@param** life

\* **@return** String with SA format for country

\*/

**public** String Country(String code, String name, String continent, **int**

area, **long** population, **float** life) {

**if** (name.length() > 18)

name = name.substring(0, 18);

DecimalFormat myFormatter = **new** DecimalFormat("#,###");

**return** code + " "+ String.*format*("%-16s %-13s %10s %13s %4.1f",

name, continent, myFormatter.format(area),

myFormatter.format(population), life);

}

/\*\*

\* Preformatted string

\* **@param** index

\* **@param** code

\* **@param** name

\* **@param** continent

\* **@param** area

\* **@param** population

\* **@param** life

\* **@return** String with Snapshot format for country

\*/

**public** String Country2(**short** ID, String code, String name,

String continent, **int** area, **long** population, **float** life) {

DecimalFormat noFormat = **new** DecimalFormat("#000");

**return** "[" + noFormat.format(ID) + "] "

+ Country(code, name, continent, area, population,

life);

}

/\*\*

\* Preformatted string

\* **@param** action

\* **@return** confirmation message string

\*/

**public** String Msg(String action){

**return** " OK, country "+action+".";

}

/\*\*

\* Preformatted string

\* **@return** invalid entry string

\*/

**public** String Sorry() {

**return** " SORRY, no country with that id.";

}

**public** String Sorry2() {

**return** " SORRY, another country has that id.";

}

**public** String Sorry3() {

**return** " SORRY, invalid id.";

}

/\*\*

\* Preformatted string

\* **@return** SA header string

\*/

**public** String Header1() {

**return** "CDE NAME------------ CONTINENT---- ------AREA ---

POPULATION LIFE";

}

/\*\*

\* Preformatted string

\* **@return** Snapshot header string

\*/

**public** String Header2(){

**return** "[ID ] "+Header1();

}

/\*\*

\* Preformatted string

\* **@return** footer

\*/

**public** String Footer() {

**return** "+++++++++++++++++++++++++++++++++

+++++++++++++++++++++++++++++++";

}

/\*\*

\* Preformatted string

\* **@return** footer

\*/

**public** String Footer2(){

**return** "++++++";

}

/\*\*

\* Writes string to file

\* **@param** toPrint string to write to file

\* **@throws** IOException

\*/

**public** **void** displayThis(String toPrint) **throws** IOException{

theLog.println(toPrint);

}

/\*\*

\* Creates log entry and closes file

\* **@throws** IOException

\*/

**public** **void** FinishUp() **throws** IOException{

theLog.println("FILE STATUS > TheLog FILE closed.");

theLog.close();

}

}

# FILE STATUS > TheLog FILE opened.

CODE STATUS > Setup started.

FILE STATUS > CountryData FILE opened.

FILE STATUS > RawData FILE opened.

FILE STATUS > RawData FILE closed.

CODE STATUS > Snapshot started.

N: 25

[ID ] CDE NAME------------ CONTINENT---- ------AREA ---POPULATION LIFE

[001] GRC Greece Europe 131,626 10,545,700 78.4

[002] EGY Egypt Africa 1,001,449 68,470,000 63.3

[003] EMPTY

[004] EMPTY

[005] CHN China Asia 9,572,900 1,277,558,000 71.4

[006] EMPTY

[007] AUS Australia Oceania 7,741,220 18,886,000 79.8

[008] USA United States North America 9,363,520 278,357,000 77.1

[009] YUG Yugoslavia Europe 102,173 10,640,000 72.4

[010] BRA Brazil South America 8,547,403 170,115,000 62.9

[011] TUR Turkey Asia 774,815 66,591,000 71.0

[012] EMPTY

[013] PSE Palestine Asia 6,257 3,101,000 71.4

[014] MEX Mexico North America 1,958,201 98,881,000 71.5

[015] HUN Hungary Europe 93,030 10,043,200 71.4

[016] ZWE Zimbabwe Africa 390,757 11,669,000 37.8

[017] WLF Wallis and Futun Oceania 200 15,000 0.0

[018] EMPTY

[019] FRA France Europe 551,500 59,225,700 78.8

[020] EMPTY

[021] QAT Qatar Asia 11,000 599,000 72.4

[022] OMN Oman Asia 309,500 2,542,000 71.8

[023] EMPTY

[024] DOM Dominican Republ North America 48,511 8,495,000 73.2

[025] NGA Nigeria Africa 923,768 111,506,000 51.6

[026] JOR Jordan Asia 88,946 5,083,000 77.4

[027] EMPTY

[028] KEN Kenya Africa 580,367 30,080,000 48.0

[029] VEN Venezuela South America 912,050 24,170,000 73.1

[030] RUS Russian Federati Europe 17,075,400 146,934,000 67.2

[031] SWE Sweden Europe 449,964 8,861,400 79.6

[032] EMPTY

[033] EMPTY

[034] LIE Liechtenstein Europe 160 32,300 78.8

[035] IND India Asia 3,287,263 1,013,662,000 62.5

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

CODE STATUS > Snapshot finished - 25 countries displayed.

FILE STATUS > CountryData FILE closed.

CODE STATUS > Setup finished - 0 countries processed.

FILE STATUS > TheLog FILE closed.

FILE STATUS > TheLog FILE opened.

CODE STATUS > UserApp started.

FILE STATUS > CountryData FILE opened.

FILE STATUS > TransData FILE opened.

SA

CDE NAME------------ CONTINENT---- ------AREA ---POPULATION LIFE

GRC Greece Europe 131,626 10,545,700 78.4

GRC Greece Europe 131,626 10,545,700 78.4

EGY Egypt Africa 1,001,449 68,470,000 63.3

CHN China Asia 9,572,900 1,277,558,000 71.4

AUS Australia Oceania 7,741,220 18,886,000 79.8

USA United States North America 9,363,520 278,357,000 77.1

YUG Yugoslavia Europe 102,173 10,640,000 72.4

BRA Brazil South America 8,547,403 170,115,000 62.9

TUR Turkey Asia 774,815 66,591,000 71.0

PSE Palestine Asia 6,257 3,101,000 71.4

MEX Mexico North America 1,958,201 98,881,000 71.5

HUN Hungary Europe 93,030 10,043,200 71.4

ZWE Zimbabwe Africa 390,757 11,669,000 37.8

WLF Wallis and Futun Oceania 200 15,000 0.0

FRA France Europe 551,500 59,225,700 78.8

QAT Qatar Asia 11,000 599,000 72.4

OMN Oman Asia 309,500 2,542,000 71.8

DOM Dominican Republ North America 48,511 8,495,000 73.2

NGA Nigeria Africa 923,768 111,506,000 51.6

JOR Jordan Asia 88,946 5,083,000 77.4

KEN Kenya Africa 580,367 30,080,000 48.0

VEN Venezuela South America 912,050 24,170,000 73.1

RUS Russian Federati Europe 17,075,400 146,934,000 67.2

SWE Sweden Europe 449,964 8,861,400 79.6

LIE Liechtenstein Europe 160 32,300 78.8

IND India Asia 3,287,263 1,013,662,000 62.5

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

SI 35

IND India Asia 3,287,263 1,013,662,000 62.5

SI 27

SORRY, no country with that id.

SI 1

GRC Greece Europe 131,626 10,545,700 78.4

SI ERROR

SORRY, invalid id.

SI 10

BRA Brazil South America 8,547,403 170,115,000 62.9

SI 33

SORRY, no country with that id.

SI 36

SORRY, no country with that id.

SI 0

SORRY, no country with that id.

SI 1

GRC Greece Europe 131,626 10,545,700 78.4

SI 99

SORRY, no country with that id.

SI 24

DOM Dominican Republ North America 48,511 8,495,000 73.2

SA

CDE NAME------------ CONTINENT---- ------AREA ---POPULATION LIFE

GRC Greece Europe 131,626 10,545,700 78.4

GRC Greece Europe 131,626 10,545,700 78.4

EGY Egypt Africa 1,001,449 68,470,000 63.3

CHN China Asia 9,572,900 1,277,558,000 71.4

AUS Australia Oceania 7,741,220 18,886,000 79.8

USA United States North America 9,363,520 278,357,000 77.1

YUG Yugoslavia Europe 102,173 10,640,000 72.4

BRA Brazil South America 8,547,403 170,115,000 62.9

TUR Turkey Asia 774,815 66,591,000 71.0

PSE Palestine Asia 6,257 3,101,000 71.4

MEX Mexico North America 1,958,201 98,881,000 71.5

HUN Hungary Europe 93,030 10,043,200 71.4

ZWE Zimbabwe Africa 390,757 11,669,000 37.8

WLF Wallis and Futun Oceania 200 15,000 0.0

FRA France Europe 551,500 59,225,700 78.8

QAT Qatar Asia 11,000 599,000 72.4

OMN Oman Asia 309,500 2,542,000 71.8

DOM Dominican Republ North America 48,511 8,495,000 73.2

NGA Nigeria Africa 923,768 111,506,000 51.6

JOR Jordan Asia 88,946 5,083,000 77.4

KEN Kenya Africa 580,367 30,080,000 48.0

VEN Venezuela South America 912,050 24,170,000 73.1

RUS Russian Federati Europe 17,075,400 146,934,000 67.2

SWE Sweden Europe 449,964 8,861,400 79.6

LIE Liechtenstein Europe 160 32,300 78.8

IND India Asia 3,287,263 1,013,662,000 62.5

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

FILE STATUS > TransData FILE closed.

CODE STATUS > Snapshot started.

N: 25

[ID ] CDE NAME------------ CONTINENT---- ------AREA ---POPULATION LIFE

[001] GRC Greece Europe 131,626 10,545,700 78.4

[002] EGY Egypt Africa 1,001,449 68,470,000 63.3

[003] EMPTY

[004] EMPTY

[005] CHN China Asia 9,572,900 1,277,558,000 71.4

[006] EMPTY

[007] AUS Australia Oceania 7,741,220 18,886,000 79.8

[008] USA United States North America 9,363,520 278,357,000 77.1

[009] YUG Yugoslavia Europe 102,173 10,640,000 72.4

[010] BRA Brazil South America 8,547,403 170,115,000 62.9

[011] TUR Turkey Asia 774,815 66,591,000 71.0

[012] EMPTY

[013] PSE Palestine Asia 6,257 3,101,000 71.4

[014] MEX Mexico North America 1,958,201 98,881,000 71.5

[015] HUN Hungary Europe 93,030 10,043,200 71.4

[016] ZWE Zimbabwe Africa 390,757 11,669,000 37.8

[017] WLF Wallis and Futun Oceania 200 15,000 0.0

[018] EMPTY

[019] FRA France Europe 551,500 59,225,700 78.8

[020] EMPTY

[021] QAT Qatar Asia 11,000 599,000 72.4

[022] OMN Oman Asia 309,500 2,542,000 71.8

[023] EMPTY

[024] DOM Dominican Republ North America 48,511 8,495,000 73.2

[025] NGA Nigeria Africa 923,768 111,506,000 51.6

[026] JOR Jordan Asia 88,946 5,083,000 77.4

[027] EMPTY

[028] KEN Kenya Africa 580,367 30,080,000 48.0

[029] VEN Venezuela South America 912,050 24,170,000 73.1

[030] RUS Russian Federati Europe 17,075,400 146,934,000 67.2

[031] SWE Sweden Europe 449,964 8,861,400 79.6

[032] EMPTY

[033] EMPTY

[034] LIE Liechtenstein Europe 160 32,300 78.8

[035] IND India Asia 3,287,263 1,013,662,000 62.5

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

CODE STATUS > Snapshot finished - 25 countries displayed.

FILE STATUS > CountryData FILE closed.

CODE STATUS > UserApp finished - 13 transactions processed.

FILE STATUS > TheLog FILE closed.

FILE STATUS > TheLog FILE opened.

CODE STATUS > UserApp started.

FILE STATUS > CountryData FILE opened.

FILE STATUS > TransData FILE opened.

FILE STATUS > TransData FILE closed.

CODE STATUS > Snapshot started.

N: 25

[ID ] CDE NAME------------ CONTINENT---- ------AREA ---POPULATION LIFE

[001] GRC Greece Europe 131,626 10,545,700 78.4

[002] EGY Egypt Africa 1,001,449 68,470,000 63.3

[003] EMPTY

[004] EMPTY

[005] CHN China Asia 9,572,900 1,277,558,000 71.4

[006] EMPTY

[007] AUS Australia Oceania 7,741,220 18,886,000 79.8

[008] USA United States North America 9,363,520 278,357,000 77.1

[009] YUG Yugoslavia Europe 102,173 10,640,000 72.4

[010] BRA Brazil South America 8,547,403 170,115,000 62.9

[011] TUR Turkey Asia 774,815 66,591,000 71.0

[012] EMPTY

[013] PSE Palestine Asia 6,257 3,101,000 71.4

[014] MEX Mexico North America 1,958,201 98,881,000 71.5

[015] HUN Hungary Europe 93,030 10,043,200 71.4

[016] ZWE Zimbabwe Africa 390,757 11,669,000 37.8

[017] WLF Wallis and Futun Oceania 200 15,000 0.0

[018] EMPTY

[019] FRA France Europe 551,500 59,225,700 78.8

[020] EMPTY

[021] QAT Qatar Asia 11,000 599,000 72.4

[022] OMN Oman Asia 309,500 2,542,000 71.8

[023] EMPTY

[024] DOM Dominican Republ North America 48,511 8,495,000 73.2

[025] NGA Nigeria Africa 923,768 111,506,000 51.6

[026] JOR Jordan Asia 88,946 5,083,000 77.4

[027] EMPTY

[028] KEN Kenya Africa 580,367 30,080,000 48.0

[029] VEN Venezuela South America 912,050 24,170,000 73.1

[030] RUS Russian Federati Europe 17,075,400 146,934,000 67.2

[031] SWE Sweden Europe 449,964 8,861,400 79.6

[032] EMPTY

[033] EMPTY

[034] LIE Liechtenstein Europe 160 32,300 78.8

[035] IND India Asia 3,287,263 1,013,662,000 62.5

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

CODE STATUS > Snapshot finished - 25 countries displayed.

FILE STATUS > CountryData FILE closed.

CODE STATUS > UserApp finished - 0 transactions processed.

FILE STATUS > TheLog FILE closed.

FILE STATUS > TheLog FILE opened.

CODE STATUS > UserApp started.

FILE STATUS > CountryData FILE opened.

FILE STATUS > TransData FILE opened.

SA

CDE NAME------------ CONTINENT---- ------AREA ---POPULATION LIFE

GRC Greece Europe 131,626 10,545,700 78.4

GRC Greece Europe 131,626 10,545,700 78.4

EGY Egypt Africa 1,001,449 68,470,000 63.3

CHN China Asia 9,572,900 1,277,558,000 71.4

AUS Australia Oceania 7,741,220 18,886,000 79.8

USA United States North America 9,363,520 278,357,000 77.1

YUG Yugoslavia Europe 102,173 10,640,000 72.4

BRA Brazil South America 8,547,403 170,115,000 62.9

TUR Turkey Asia 774,815 66,591,000 71.0

PSE Palestine Asia 6,257 3,101,000 71.4

MEX Mexico North America 1,958,201 98,881,000 71.5

HUN Hungary Europe 93,030 10,043,200 71.4

ZWE Zimbabwe Africa 390,757 11,669,000 37.8

WLF Wallis and Futun Oceania 200 15,000 0.0

FRA France Europe 551,500 59,225,700 78.8

QAT Qatar Asia 11,000 599,000 72.4

OMN Oman Asia 309,500 2,542,000 71.8

DOM Dominican Republ North America 48,511 8,495,000 73.2

NGA Nigeria Africa 923,768 111,506,000 51.6

JOR Jordan Asia 88,946 5,083,000 77.4

KEN Kenya Africa 580,367 30,080,000 48.0

VEN Venezuela South America 912,050 24,170,000 73.1

RUS Russian Federati Europe 17,075,400 146,934,000 67.2

SWE Sweden Europe 449,964 8,861,400 79.6

LIE Liechtenstein Europe 160 32,300 78.8

IND India Asia 3,287,263 1,013,662,000 62.5

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

SI 8

USA United States North America 9,363,520 278,357,000 77.1

SI 23

SORRY, no country with that id.

DI 15

OK, country deleted.

DI 5

OK, country deleted.

DI 14

OK, country deleted.

DI 15

SORRY, no country with that id.

SI 5

SORRY, no country with that id.

SI 14

SORRY, no country with that id.

SI 15

SORRY, no country with that id.

DI 15

SORRY, no country with that id.

DI 4

SORRY, no country with that id.

DI 27

SORRY, no country with that id.

DI 0

SORRY, no country with that id.

DI ERROR

SORRY, invalid id.

IN 4,ARG,Argentina,2780400,37032000,75.1

OK, country inserted.

SI 4

ARG Argentina South America 2,780,400 37,032,000 75.1

IN 37,JPN,Japan,377829,126714000,80.7

OK, country inserted.

IN 27,JAM,Jamaica,10990,2583000,75.2

OK, country inserted.

IN 15,CUB,Cuba,110861,11201000,76.2

OK, country inserted.

IN 5,AUT,Austria,83859,8091800,77.7

OK, country inserted.

SI 15

CUB Cuba North America 110,861 11,201,000 76.2

DI 15

OK, country deleted.

DI 27

OK, country deleted.

DI 1

OK, country deleted.

DI 35

OK, country deleted.

IN 15,SGS,South Georgia and the South Sandwich Islands,3903,0,0.0

OK, country inserted.

IN 20,HMD,Heard Island and McDonald Islands,359,0,0.0

OK, country inserted.

IN 40,DEU,Germany,357022,82164700,77.4

OK, country inserted.

SI 15

SGS South Georgia an Antarctica 3,903 0 0.0

SI 35

SORRY, no country with that id.

SI 1

SORRY, no country with that id.

SI 20

HMD Heard Island and Antarctica 359 0 0.0

SI 40

DEU Germany Europe 357,022 82,164,700 77.4

SI 5

AUT Austria Europe 83,859 8,091,800 77.7

SI 37

JPN Japan Asia 377,829 126,714,000 80.7

SI 27

SORRY, no country with that id.

SI 36

SORRY, no country with that id.

SI 99

SORRY, no country with that id.

SI 41

SORRY, no country with that id.

SI 0

SORRY, no country with that id.

SI 30

RUS Russian Federati Europe 17,075,400 146,934,000 67.2

SI 28

KEN Kenya Africa 580,367 30,080,000 48.0

SA

CDE NAME------------ CONTINENT---- ------AREA ---POPULATION LIFE

EGY Egypt Africa 1,001,449 68,470,000 63.3

ARG Argentina South America 2,780,400 37,032,000 75.1

AUT Austria Europe 83,859 8,091,800 77.7

AUS Australia Oceania 7,741,220 18,886,000 79.8

USA United States North America 9,363,520 278,357,000 77.1

YUG Yugoslavia Europe 102,173 10,640,000 72.4

BRA Brazil South America 8,547,403 170,115,000 62.9

TUR Turkey Asia 774,815 66,591,000 71.0

PSE Palestine Asia 6,257 3,101,000 71.4

SGS South Georgia an Antarctica 3,903 0 0.0

ZWE Zimbabwe Africa 390,757 11,669,000 37.8

WLF Wallis and Futun Oceania 200 15,000 0.0

FRA France Europe 551,500 59,225,700 78.8

HMD Heard Island and Antarctica 359 0 0.0

QAT Qatar Asia 11,000 599,000 72.4

OMN Oman Asia 309,500 2,542,000 71.8

DOM Dominican Republ North America 48,511 8,495,000 73.2

NGA Nigeria Africa 923,768 111,506,000 51.6

JOR Jordan Asia 88,946 5,083,000 77.4

KEN Kenya Africa 580,367 30,080,000 48.0

VEN Venezuela South America 912,050 24,170,000 73.1

RUS Russian Federati Europe 17,075,400 146,934,000 67.2

SWE Sweden Europe 449,964 8,861,400 79.6

LIE Liechtenstein Europe 160 32,300 78.8

JPN Japan Asia 377,829 126,714,000 80.7

DEU Germany Europe 357,022 82,164,700 77.4

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

FILE STATUS > TransData FILE closed.

CODE STATUS > Snapshot started.

N: 26

[ID ] CDE NAME------------ CONTINENT---- ------AREA ---POPULATION LIFE

[001] EMPTY

[002] EGY Egypt Africa 1,001,449 68,470,000 63.3

[003] EMPTY

[004] ARG Argentina South America 2,780,400 37,032,000 75.1

[005] AUT Austria Europe 83,859 8,091,800 77.7

[006] EMPTY

[007] AUS Australia Oceania 7,741,220 18,886,000 79.8

[008] USA United States North America 9,363,520 278,357,000 77.1

[009] YUG Yugoslavia Europe 102,173 10,640,000 72.4

[010] BRA Brazil South America 8,547,403 170,115,000 62.9

[011] TUR Turkey Asia 774,815 66,591,000 71.0

[012] EMPTY

[013] PSE Palestine Asia 6,257 3,101,000 71.4

[014] EMPTY

[015] SGS South Georgia an Antarctica 3,903 0 0.0

[016] ZWE Zimbabwe Africa 390,757 11,669,000 37.8

[017] WLF Wallis and Futun Oceania 200 15,000 0.0

[018] EMPTY

[019] FRA France Europe 551,500 59,225,700 78.8

[020] HMD Heard Island and Antarctica 359 0 0.0

[021] QAT Qatar Asia 11,000 599,000 72.4

[022] OMN Oman Asia 309,500 2,542,000 71.8

[023] EMPTY

[024] DOM Dominican Republ North America 48,511 8,495,000 73.2

[025] NGA Nigeria Africa 923,768 111,506,000 51.6

[026] JOR Jordan Asia 88,946 5,083,000 77.4

[027] EMPTY

[028] KEN Kenya Africa 580,367 30,080,000 48.0

[029] VEN Venezuela South America 912,050 24,170,000 73.1

[030] RUS Russian Federati Europe 17,075,400 146,934,000 67.2

[031] SWE Sweden Europe 449,964 8,861,400 79.6

[032] EMPTY

[033] EMPTY

[034] LIE Liechtenstein Europe 160 32,300 78.8

[035] EMPTY

[036] EMPTY

[037] JPN Japan Asia 377,829 126,714,000 80.7

[038] EMPTY

[039] EMPTY

[040] DEU Germany Europe 357,022 82,164,700 77.4

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

CODE STATUS > Snapshot finished - 26 countries displayed.

FILE STATUS > CountryData FILE closed.

CODE STATUS > UserApp finished - 44 transactions processed.

FILE STATUS > TheLog FILE closed.