Overview

The MSDRGMF381 software is provided to users to allows them to group MS-DRG claims using the V38.1 software in a Java environment, either as a standalone executable Jar file or embedded within a calling program.

Setup

The following files are needed to execute the MsDrgMF381 software:

1. MSDRGMF381.jar – the jar file to be run, vie either standalone mode or by calls from an outside program.

Interface Methods

Users wishing to embed this jar in their own applications may do so by invoking the following method:

String results = grouper.process(inputRecord)

The method takes a String in the format specified in Table 1 and returns the grouping results as a String in the format specified in Table 2.

An example program that calls the grouper might look something like this:

public void myProgram() {

String inputRecord = null;

Msdrg grouper = new Msdrg();

// populate the inputRecord string

String results = grouper.processMsdrg(inputRecord);

// do something with the results

}

Input & Output Formats

The total length of the input record is **638** bytes. The expected format of the input for this program is outlined in the following table:

Table 1. Input Format

|  |  |
| --- | --- |
| **Field** | **Length** |
| CLAIM-AGE | PIC X(03) |
| CLAIM-SEX | PIC X(01) |
| CLAIM-DISCHARGE-STATUS | PIC X(02) |
| CLAIM-POA | PIC X(01) |
| CLAIM-ADMIT-DATE | PIC X(08) |
| CLAIM-DISCHARGE-DATE | PIC X(08) |
| DIAG\_CODES | PIC X(200) |
| PROC-CODES | PIC X(175) |
| CLAIM-PROCDATES | PIC X(200) |
| CLAIM-OPT-DATA | PIC X(40) |

The total length of the input record is **1162** bytes. The expected format of the output from this program is outlined in the following table:

Table 2. Output Format

|  |  |
| --- | --- |
| **Field** | **Length** |
| OUT-CMSDRG-RTC | PIC X(02) |
| OUT-CMSDRG-MDC | PIC X(02) |
| OUT-CMSDRG-DRG | PIC X(04) |
| OUT-CMSDRG-OUTGRFLAGS | PIC X(05) |
| OUT-CMSDRG-OUTDXFLAGS | PIC X(625) |
| OUT-CMSDRG-OUTPRFLAGS | PIC X(500) |
| OUT-CMSDRG-OUTBUFF | PIC X(24) |

|  |
| --- |
| **Media contents** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **File** | **File name** | **LRECL** | **BLKSIZE** | **Description** | |  |  |  |  |  | | 1 | OBJLIB | 80 | 27920 | Object library | | 2 | SRCLIB | 80 | 32720 | Source library | | 3 | LOADLIB | 0 | 6233 | Load library | |  |  |  |  |  | | 4 | TESTDB | 2000 | 18000 | Test Database | | 5 | JCL | 80 | 27920 | Sample JCL | |  |  |  |  |  | |

**Grouper program installation**

All required software for executing the MS-DRG Java grouper is contained in the folders in this directory.

This directory contains the following folders:

• Load library - MS-DRG grouper load modules

• Object library - MS-DRG grouper object modules

• Source library - MS-DRG grouper source programs

Test database file

Sample JCL

* Java Jar
* Environment file

JCL Library

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * Sample JCL library members | | | |  |
| Number | Name | Description | |
| 1 | BUILDPDS | Sample JCL used for electronic download | |
| 2 | COBTSTGO | Run test database, executing COBTEST load library members | |
| 3 | COBTEST | Run sample COBOL program (COBTEST) | |
| 4 | INSTLCNT | Readme for install test database record counts | |

The following steps download the JCL library.

1. Allocate a PDSE on your mainframe with the following characteristics:

DSN = [e.g. YOURID.GROUPER.JCL]

RECFM = FB

LRECL = 80

BLKSIZE = 27920

SPACE = (TRK(2,1,4),RLSE)

2. FTP in ASCII mode all the files in the sample JCL library folder into the PDSE allocated in step 1.

*Load library*

The load library is a sequential file, FTPLOAD.

The load library consists of the load modules for the MS-DRG Grouper.

1. Pre-allocate a sequential dataset on your mainframe to receive the file using the following file characteristics:

DSN = [e.g. YOURID.GROUPER.FTPLOAD]

RECFM = FB

LRECL = 80

BLKSIZE = 3120

SPACE = (CYL(1,1),RLSE)

2. FTP in BINARY mode the FTPLOAD file into the sequential dataset you allocated above.

**Important!** You must FTP the load module files in BINARY.

3. Pre-allocate a load library PDSE on the mainframe using the following file characteristics:

DSN = [e.g. YOURID.GROUPER.LOADLIB]

RECFM = U

BLKSIZE = 6233

SPACE = (CYL(1,3,2),RLSE)

4. Create a BUILDPDS JCL member as follows:

Add your JOBCARD

Modify dataset names as necessary

INDATASET = sequential dataset that was FTP’d to the mainframe in the step above.

DATASET = pre-allocated load library PDSE(Library) that was created in the step above.

**Note:** This JCL executes the utility, IKJEFT01, a terminal monitor program that executes the TSO commands via batch processing. This will populate the LOAD LIBRARY from the FTP’d load sequential file. A copy is included in the JCLLIB folder

After you modify the BUILDPDS, execute the JCL.

**Load Library Contents**

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Description** |
| 1 | DRG381JV | Java wrapper Control program |
| 2 | COBTEST | Sample COBOL interface program |
| 3 | DRG381RN | Sample program to execute files |

*Object library*

This information is for the object library. This directory contains an object module folder.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Object library contents** | | | |  | |
| **Number** | **Name** | **Description** | |
| 1 | DRG381JV | Java wrapper Control program | |
| 2 | COBTEST | Sample COBOL interface program | |
| 3 | DRG381RN | Sample program to execute files | |

**Important!** Object module files must be FTP’d in BINARY.

The following steps download the object library.

1. Allocate a PDSE on your mainframe with the following characteristics:

DSN = [e.g. YOURID.GROUPER.OBJLIB]

RECFM = FB

LRECL = 80

BLKSIZE = 27920

SPACE = (CYL(1,1,2),RLSE)

1. FTP in **BINARY mode** all the files in the object library folder into the PDSE allocated in step 1.

*Source library*

There are several datasets included on the media that are not needed for the grouping process but may be useful to grouper users.

The folder contains the source library for all the grouper programs, tables, and the COBOL test programs. The library contains three members, as listed in the following table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Source library contents** | | | |  |
| **umber** | **Name** | **Description** | |
| 1 | DRG381JV | Java wrapper Control program | |
| 2 | COBTEST | Sample COBOL interface program | |
| 3 | DRG381RN | Sample program to execute files | |

The following steps are required to FTP the source library to the mainframe.

1. Allocate a PDSE on your mainframe with the following characteristics:

DSN = [e.g. YOURID.GROUPER.SRCLIB]

RECFM = FB

LRECL = 80

BLKSIZE = 32720

SPACE = (CYL(1,1,4),RLSE)

2. FTP in ASCII mode all the files in the source library folder into the PDSE allocated in step 1.

*Test Database File*

The following steps load the test database file to the mainframe.

1. Allocate a sequential file (PS) on your mainframe using the attributes below.

DSN=YOURID.GROUPER.**TESTDB**

RECFM=FB

LRECL=2000

BLKSIZE=18000

SPACE=(CYL,(20,1),RLSE)

2.FTP the TESTDB file in ASCII mode from the testdb folder to the mainframe, YOURID.GROUPER.**TESTDB**.

Layout is in sample COBTEST source program.

Java Modules, and Environment file.

On the USS system, create a folder ‘dist’ and FTP the modules in the java-jar folders in Binary mode into

That folder. One member is included: MSDRGMF381.jar file

FTP the DRGENV381 file in ascii mode and place the file at the same level as the dist folder.  
Update the above files with the ***correct path in your environment***.