Overview

The MCEMF381 software is provided to users to allows them to group MCE claims using the V38.1 software in a Java environment, embedded within a calling program.

Setup

The following files are needed to execute the MCE MF java version 38.1 software:

1. MCEMFV381.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 static method:

String results = Mce.processMce(inputRecord)

Alternatively, users could initialize the MCE object and invoke the processing logic:

Mce mceComponent = new Mce();

String results = mceComponent.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 (alternative method) that calls the editor might look something like this:

public void myProgram() {

String inputRecord = null;

Mce mceComponent = new Mce();

// populate the inputRecord string then send to .process()

String results = mceComponent.process(inputRecord);

// do something with the results

}

Input & Output Formats to/from jar and to/from OO Cobol wrapper program MCT381JV

The total length of the input record is **500** bytes. The expected format of the input for this program is outlined in the following table:

Table . Input format to invoke jar and to call OO Cobol wrapper program MCT381JV

|  |  |
| --- | --- |
| **Field** | **Length** |
| CLAIM-AGE | PIC X(03) |
| CLAIM-SEX | PIC X(01) |
| CLAIM-DISCHARGE-STATUS | PIC X(02) |
| CLAIM-LOS | PIC X(05) |
| CLAIM-DISCHARGE-DATE | PIC X(08) |
| DIAG-CODES | PIC X(208). First 8 characters are for ADX and POA |
| PROC-CODES | PIC X(175) |
| WS-PROVIDER-NUM | PIC X(15) |
| WS-PPS | PIC X(01) |
| WS-I9-I10-IND | PIC X(01) 9=I9, 0=I10 |
| FILLER | PIC X(80) |
| WS-DEBUG-IND (Internal field) | PIC X(01) |

The total length of the input record is **917** bytes. The expected format of the output from this program is outlined in the following table:

Table . Output format from jar and OO Cobol wrapper program MCT381JV

|  |  |
| --- | --- |
| **Field** | **Length** |
| OUT-CMSMCE-VER | PIC X(03) |
| OUT-CMSMCE-OUTADXFLAG | PIC X(01) |
| OUT-CMSMCE-OUTDXFLAGS | PIC X(350) |
| OUT-CMSMCE-OUTPRFLAGS | PIC X(425) |
| OUT-CMSMCE-OUTBUFF | PIC X(138) |

|  |
| --- |
| **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 | 1400 | 18200 | Test Database | | 5 | JCL | 80 | 27920 | Sample JCL | |  |  |  |  |  | |

**Editor program installation**

All required software for executing the MCE MF Java editor is contained in the folders in this directory.

This directory contains the following folders:

• Load library - MCE load modules

• Object library - MCE object modules

• Source library - MCE source programs

Test database file – Same file as MCE V38.1 January 1, 2021 release can be used for this test.

Sample JCL

* The VSAM description file from the MCE Version 38.1 January 1, 2021 release can be used for this test.
* Java Jar
* Environment file

JCL Library

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| * **Sample JCL library members** | | | |  |
| **Number** | **Name** | **Description** | |
| 1 | BUILDPDS | Sample JCL used for electronic download | |
| 2 | CBTSTGO | Run test database, executing COBTEST load library members | |
| 3 | COBTEST | Run sample COBOL program (COBTEST) | |

The following steps download the JCL library.

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

DSN = [e.g. YOURID.EDITOR.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 MCE Java programs.

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

DSN = [e.g. YOURID.EDITOR.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.EDITOR.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 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 | MCT381JV | Cobol Java wrapper Control program |
| 2 | COBTEST | Sample Cobol interface program |
| 3 | MCT381PA | Print program |

*Object library*

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Object library contents** | | | |  | |
| **Number** | **Name** | **Description** | |
| 1 | MCT381JV | Cobol Java wrapper control program | |
| 2 | COBTEST | Sample Cobol interface program | |
| 3 | MCT381PA | Print Program | |
| 4 | MCT381DT | Date Calculation Program | |
| 5 | MCT381VS | VSAM code description program | |

**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.EDITOR.OBJLIB]

RECFM = FB

LRECL = 80

BLKSIZE = 27920

SPACE = (CYL(1,1,2),RLSE)

2.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 editor users.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Source library contents** | | | |  |
| **Number** | **Name** | **Description** | |
| 1 | MCT381JV | Java wrapper control program | |
| 2 | COBTEST | Sample COBOL interface program | |

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.EDITOR.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.EDITOR.**TESTDB**

RECFM=FB

LRECL=1400

BLKSIZE=18200

SPACE=(CYL,(18,1),RLSE)

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

Layout is in sample COBTEST source program.

Java Modules, Environment and properties files.

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: MCEMFV381.jar file

FTP the MCEENV381 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.