

Medicare Severity Diagnosis Related Groups (MS-DRG) Software

# **Software Installation Guide for z/OS Batch**

ICD-10-CM

v37.0 R1 October 2019

PBL-036  
October 2019

# Table of Contents

<b>Preface</b> .....	<b>vii</b>
<b>Chapter 1: Introduction</b> .....	<b>9</b>
Data format requirements .....	9
Information returned by the grouper .....	12
Grouper return code .....	14
Flags returned by the grouper .....	14
Ancillary buffer .....	19
<b>Chapter 2: Installing the MS-DRG Software</b> .....	<b>21</b>
eDownload instructions .....	22
Grouper program installation .....	22
<i>Load library</i> .....	22
<i>Object library</i> .....	24
<i>Source library</i> .....	25
Miscellaneous files installation .....	26
<i>Test Database File</i> .....	26
<i>MDC Description File</i> .....	27
<i>DRG 3 Description File</i> .....	28
<i>DRG 4 Description File</i> .....	28
ICD-10-CM/PCS MS-DRG Grouper Tables .....	29
<i>Attributes</i> .....	29
<i>Procedure clusters</i> .....	30
<i>Grouper logic (drglogic)</i> .....	30
<i>Diagnosis attributes (dxattlst)</i> .....	31
<i>Diagnosis table (dxlist)</i> .....	31
<i>Diagnosis pattern table (dxpatern)</i> .....	31
<i>Procedure attributes (prattlst)</i> .....	32
<i>Procedure table (prlist)</i> .....	32
<i>Procedure pattern table (prpatern)</i> .....	33
<i>Procedure cluster definitions (prclstrs)</i> .....	33
<i>CC/MCC exclusions (exclusn)</i> .....	34
Uploading grouper tables/files to the mainframe .....	34
<i>DRG logic file</i> .....	34
<i>Diagnosis attribute list</i> .....	34
<i>Diagnosis list file</i> .....	35
<i>Diagnosis pattern list</i> .....	35
<i>CC/MCC exclusions file</i> .....	35
<i>Procedure attribute list</i> .....	36

<i>Procedure list file</i> .....	36
<i>Procedure pattern list</i> .....	37
<i>Procedure clusters file</i> .....	37

### **Chapter 3: Using and testing the grouper utility ..... 39**

Link-editing the grouper utility .....	39
Using the grouper utility .....	39
Control statement examples .....	40
<i>The discharge diagnosis control statement (DDX)</i> .....	40
<i>The procedure control statement (SRG)</i> .....	41
<i>The age control statement (AGE)</i> .....	41
<i>The sex control statement (SEX)</i> .....	42
<i>The discharge status control statement (DSP)</i> .....	42
<i>The present on admission control statement (POA)</i> .....	42
<i>The admission date control statement (ADT)</i> .....	42
<i>The discharge date control statement (DDT)</i> .....	43
<i>The procedure dates control statement (SDT)</i> .....	43
Grouper output control statements .....	43
<i>The return code control statement (RTC)</i> .....	43
<i>The MDC control statement (MDC)</i> .....	44
<i>The DRG control statement (DRG)</i> .....	44
<i>The grouper flags control statement (GFL)</i> .....	44
<i>The diagnosis flags control statement (DFL)</i> .....	44
<i>The procedure flags control statement (SFL)</i> .....	44
<i>The buffer control statement (BUF)</i> .....	45
Running the grouper utility program .....	45

### **Chapter 4: Using the grouper with higher-level languages ..... 47**

General strategy for COBOL driving program .....	47
Input to the grouper subroutines .....	48
Output from the grouper subroutines .....	50
Using the alternate interface .....	50
Executor processing of the diagnosis and procedure buffers .....	51

### **Chapter 5: The MS-DRG grouper executor ..... 53**

Construction of the record mask .....	53
DRG determination .....	54
Testing for the ONLY surgery condition .....	54
Testing for the ONLY DX condition .....	55
Testing for the OWISE condition .....	55
Testing for the ANYCOMB condition .....	55
CC exclusion subroutine .....	55
Testing for the OTHOR condition .....	56

Testing for illogical principal diagnosis .....	56
Testing for multiple significant trauma.....	56
Finding codes that affect Initial DRG assignment .....	57
Final DRG.....	57
Executor ABEND codes .....	57
<b>Appendix A: Grouping results for the test database .....</b>	<b>59</b>
<b>Index .....</b>	<b>63</b>



# Preface

This manual contains the information needed to use the Medicare Severity Diagnosis Related Groups (MS-DRG) Software (the grouper), version 37.0 R1 in a mainframe environment. Two interface versions to the MS-DRG software are supplied. One, the standard version, assumes that the operating system is z/OS Batch. The second is re-entrant and uses no macros and so can be used in a variety of operating system environments, although it requires additional parameters from the calling program.

This manual provides technical personnel with the detail necessary to install, debug, and support the MS-DRG software. The first four chapters describe installing, testing, and running the grouper. Chapter 5 provides detailed information on the logic of the executor and the construction of the tables. An appendix provides grouping results for the test database.

Users already familiar with the MS-DRG software are encouraged to read this manual to ensure that installation, testing, and production runs perform without incident. If you have never used the software, we strongly recommend that you read the manual thoroughly to become familiar with it before installation.

The manual assumes that you are familiar with:

- IBM Basic Assembler Language (BAL)
- IBM MVS Job Control Language



# Chapter 1: Introduction

This manual provides technical personnel with the detail necessary to install and understand the Medicare Severity Diagnosis Related Groups (MS-DRG) Software (the grouper) so they can install, interface with, and support it.

The MS-DRG software may be implemented either as a set of subroutines to be called from a program written in Assembler or a higher level language (e.g., COBOL) or as a utility program with all parameters passed through a job's SYSIN input stream.

For example purposes, the required datasets are copied to disk and cataloged under the Userid, GROUPER.

The current version of this software was developed and tested using z/OS version 2.1.

**Note:** The first release of this software in 2020 will be developed and tested using z/OS version 2.3.

## Data format requirements

The grouper executor is contained in three Basic Assembler Language (BAL) programs. The data formats required by the executor are shown in the table that follows.

If these data requirements are met, the grouper may be implemented by using a utility program (page [39](#)). Whenever these requirements are not met, the grouper must be implemented as a subroutine to a higher level language program that re-codes the information as necessary (page [47](#)).

**Table 1. Required data formats**

Name	Length in bytes	Description
Diagnosis	8	First 7 bytes represent the diagnosis code. Left-justified, blank-filled, up to 25 accepted. The eighth byte represents the POA indicator. Y - Yes, present at the time of inpatient admission N - No, not present at the time of inpatient admission U - Insufficient documentation to determine if present on admission W - Clinically unable to determine if present at time of admission 1 - Code is exempt from POA reporting (Used on 4010 form) Blank - Code is exempt from POA reporting (Used on 5010 form, effective 01/01/2011)
Procedure	7	Left-justified, blank-filled, up to 25 accepted
Age	3	0 (zero) through 124, right-justified
Sex	1	0-2 (0-unknown, 1-male, 2-female)
Discharge Status	2	01-Home, Self-Care 02-Short Term Hosp 03-SNF 04-Custodial/supportive care (effective 10/1/2009) 05-Canc/Child hosp 06-Home Health Service 07-Against Medical Advice 20-Died 21-Court/law enfrc 30-Still A Patient 43-FedHospital 50-Hospice-Home 51-Hospice-Medical Facility 61-Swing Bed 62-Rehab facility/rehab unit 63-Long term care hospital 64-Nursing facility - Medicaid certified 65-Psych hosp/unit

Name	Length in bytes	Description
<i>(continued)</i> Discharge Status		<i>(continued)</i> 66-Critical Access Hospital 69-Designated Disaster Alternative Care Site 70-Oth institution 81-Home-Self care w Planned Readmission 82-Short Term Hospital w Planned Readmission 83-SNF w Planned Readmission 84-Cust/supp care w Planned Readmission 85-Canc/child hosp w Planned Readmission 86-Home Health Service w Planned Readmission 87-Court/law enfrc w Planned Readmission 88-Federal Hospital w Planned Readmission 89-Swing Bed w Planned Readmission 90-Rehab Facility/ Unit w Planned Readmission 91-LTCH w Planned Readmission 92-Nursg Fac-Medicaid Cert w Planned Readmiss 93-Psych Hosp/Unit w Planned Readmission 94-Crit Acc Hosp w Planned Readmission 95-Oth Institution w Planned Readmission
POA logic	1	Present On Admission (POA) logic indicator. X - Exempt from POA reporting Z - Requires POA reporting
Admit Date	8	Format = YYYYMMDD (for use with future POA logic)
Discharge Date	8	Format = YYYYMMDD (for use with future POA logic)
Procedure Dates	200	Date of each procedure code Format = YYYYMMDD (for use with future POA logic)

## Information returned by the grouper

The information returned by the grouper is shown in the following tables.

The field DRG listed below represents the 3-digit MS-DRG number used by the Centers for Medicare & Medicaid Services (CMS) for DRG payment purposes. The 3-byte “initial DRG” field in the ancillary buffer represents the DRG prior to the application of the HAC logic. The ancillary buffer also contains 4-byte initial and final DRG numbers. These 4-byte DRG numbers are for statistical purposes only. Each 3-digit DRG concept is split on MCC, CC, and non-CC to create the 4-digit DRG.

For example, as a 3-digit DRG, Non-specific CVA & precerebral occlusion w/o infarction is split into 067 (w MCC) and 068 (w/o MCC). As a 4-digit DRG, Non-specific CVA & precerebral occlusion w/o infarction is split into 0671 (w MCC), 0672 (w CC), and 0673 (w/o CC/MCC). There are also “initial” and “final” flags in the diagnosis flag buffer. The flags indicating which secondary diagnoses and procedures affect DRG assignment apply to the 3-digit DRG, but could be different for the 4-digit DRG.

The grouper executor may be implemented as a subroutine to be called from Assembler or a higher-level language program. This chapter shows how this may be done for a COBOL programming environment. To create the subroutines, you must have copied file 1 (the grouper objlib) from the media to disk (page [21](#)).

**Table 2. Information returned by the MS-DRG software**

Name	Length in bytes	Description
RTC	2	Grouper return code (page <a href="#">14</a> ) 00 - Record grouped 01 - Diagnosis code cannot be used as principal dx 02 - Record does not meet criteria for any DRG in the MDC that is indicated by principal dx 03 - Invalid age 04 - Invalid sex 05 - Invalid discharge status 06 - Illogical principal diagnosis 07 - Invalid principal diagnosis 09 - POA logic indicator = Z and at least one HAC POA is invalid, missing, or 1 10 - POA logic indicator is invalid or missing and at least one HAC POA is N or U 11 - POA logic indicator is missing or invalid, and at least one HAC POA is invalid, missing, or 1 12 - (not valid effective 10/1/2010) POA logic indicator = Z and at least one HAC POA =1 13 - (not valid effective 10/1/2010) POA logic indicator is invalid or missing and at least one HAC POA = 1 14 - (not valid effective 10/1/2010) POA logic indicator = Z and there are multiple HACs that have different HAC POA values that are not Y, W, N, U 15 - POA logic indicator is missing or invalid, and there are multiple HACs that have different HAC POA values that are not Y or W
Final MDC	2	Major Diagnostic Category number (00 - 25) assigned to patient record
Final DRG	4	Diagnosis Related Group number (0001 - 0999) assigned to patient record (after HAC logic is applied)
GRFLGS	5	See table for "Grouper flags returned by the MS-DRG software" (page <a href="#">15</a> )
DXFLGS	625	See table for "Diagnosis flags returned by the grouper" (page <a href="#">15</a> )

Name	Length in bytes	Description
PRFLGS	500	See table for "Procedure flags returned by the grouper" (page <a href="#">17</a> )

## Grouper return code

The grouper return code (RTC) indicates whether or not the grouping process was successful for a given record. The following table describes the values for the Return Code.

**Table 3. Return code descriptions**

Return code	Description
1	The first listed diagnosis is a valid code but it can not be used as principal diagnosis. An example of this situation would be any one of the ICD-10-CM "V" codes, which are not indicative of the MDC into which this patient should be classified.
2	This code occurs when all of the DRG criteria for the MDC have been examined and the record does not match any of them.
3,4 and 5	These codes occur only for those DRGs that are part of grouping criteria (i.e., the grouper does not perform an automatic edit check of age, sex, and discharge status).
6	The principal diagnosis is considered illogical, meaning that it is unlikely that there would be an occurrence.
7	The code used as principal diagnosis is not a valid ICD-10-CM code.
9, 10, 11, 15	These codes occur when there is at least one HAC on the record and there is an issue with either the POA logic indicator or the POA values assigned to the HAC.

## Flags returned by the grouper

The following tables show the information returned by the grouper regarding DRGs, diagnoses, and procedures.

Some secondary diagnosis codes can be assigned to multiple Hospital Acquired Conditions (HACs). In the diagnosis flags table, the fields for Hospital Acquired Condition (HAC) assignment criteria and Hospital Acquired Condition (HAC) Usage have been expanded to five occurrences.

This allows for the capture of all HACs met by a secondary diagnosis. In the procedure flags table, the Hospital Acquired Condition (HAC) assignment criteria field has been expanded to five occurrences to reflect all HACs the procedure was used in, along with the secondary diagnosis that satisfied the HAC criteria.

**Table 4. Grouper flags returned by the MS-DRG software**

Position	Description
1 and 2	Number of unique Hospital Acquired Conditions (HAC) met
3	Final CC/MCC impact on DRG assignment: 0 = DRG assigned is not based on the presence of a CC or MCC 1 = DRG assigned is based on presence of MCC 2 = DRG assigned is based on presence of CC
4	Initial CC/MCC impact on DRG assignment: 0 = DRG assigned is not based on the presence of a CC or MCC 1 = DRG assigned is based on presence of MCC 2 = DRG assigned is based on presence of CC
5	HAC Status 0 = HAC Not Applicable; Hospital is exempt or HAC criteria not met 1 = Criteria for one or more HACs met, Final DRG did not change 2 = Criteria for one or more HACs met, Final DRG changed 3 = Criteria for one or more HACs met, Final DRG changed to 999

**Table 5. Diagnosis flags returned by the grouper**

Position	Description (25 characters per diagnosis)
1	0 = Diagnosis invalid 1 = Diagnosis valid
2	Diagnosis affects DRG 0 = Diagnosis not used to assign DRG 1 = Diagnosis affected the initial DRG only 2 = Diagnosis affected the final DRG only 3 = Diagnosis affected both initial and final DRG

Position	Description (25 characters per diagnosis)
3	CC/MCC Categorization 0 = Diagnosis is not considered a Major CC or CC for this patient 1 = Diagnosis is a Major CC for both initial and final DRG 2 = Diagnosis is a CC for both initial and final DRG 3 = Diagnosis is a MCC for initial DRG and a Non-CC for final DRG 4 = Diagnosis is a CC for initial DRG and a Non-CC for final DRG 5 = Diagnosis is a MCC but not considered due to PDX/SDX exclusion 6 = Diagnosis is a CC but not considered due to PDX/SDX exclusion
4 and 5	Hospital Acquired Condition (HAC) assignment criteria #1 00 = Criteria to be assigned as an HAC not met 01 = Foreign Object Retained After Surgery 02 = Air Embolism 03 = Blood Incompatibility 04 = Stage III and IV pressure ulcers 05 = Falls and Trauma 06 = Catheter Associated UTI 07 = Vascular Catheter-Associated Infection 08 = Infection following CABG 09 = Manifestations of poor glycemic control 10 = DVT/PE following knee or hip replacement 11 = Infection following bariatric surgery 12 = Infection following certain orthopedic procedures of spine, shoulder or elbow 13 = Surgical site infection (SSI) following cardiac implantable electronic device (CIED) procedures 14 = Iatrogenic pneumothorax w/ venous catheterization
6	Hospital Acquired Condition (HAC) Usage #1 0 = Dx not on HAC list, not applicable 1 = Dx on HAC list and HAC criteria met 2 = Dx on HAC list and HAC criteria not met 3 = Dx on HAC list, but HAC not applicable due to PDX/SDX exclusion 4 = HAC not applicable, hospital is exempt from POA reporting
7 and 8	Hospital Acquired Condition (HAC) assignment criteria #2 Refer to positions 4 and 5 for a list of values.

<b>Position</b>	<b>Description (25 characters per diagnosis)</b>
9	Hospital Acquired Condition (HAC) Usage #2 Refer to position 6 for a list of values.
10 and 11	Hospital Acquired Condition (HAC) assignment criteria #3 Refer to positions 4 and 5 for a list of values.
12	Hospital Acquired Condition (HAC) Usage #3 Refer to position 6 for a list of values.
13 and 14	Hospital Acquired Condition (HAC) assignment criteria #4 Refer to positions 4 and 5 for a list of values.
15	Hospital Acquired Condition (HAC) Usage #4 Refer to position 6 for a list of values.
16 and 17	Hospital Acquired Condition (HAC) assignment criteria #5 Refer to positions 4 and 5 for a list of values.
18	Hospital Acquired Condition (HAC) Usage #5 Refer to position 6 for a list of values.
19-25	Filler

**Table 6. Procedure flags returned by the grouper**

Position	Description (20 characters per procedure)
1	0 = Procedure invalid 1 = Procedure valid
2	Procedure affects DRG* 0 = Procedure did not affect DRG assignment 1 = Procedure affected the initial DRG assignment only 2 = Procedure affected the final DRG assignment only 3 = Procedure affected both initial and final DRG assignment * When there are two or more procedures on the record that could impact either the initial, final or both DRG assignments: If one of these procedures is in the first procedure position, that procedure will be be flagged as 1, 2 or 3 with the following exceptions: a. If a single procedure designating a complete system is tied with a combination pair that also designated a complete system, the single procedure will be flagged regardless of position. b. If multiple combinations of lead/device pairs are tied then only one pair will be flagged regardless of position. c. If the two procedures tied are an OR and non-OR, the OR will be flagged regardless of position. If none of the tied procedures is in the first procedure position, then the procedure with the lowest ascii/index value will be flagged as 1, 2 or 3.
3	0 = Procedure is not an OR procedure 1 = Procedure is an OR procedure
4 and 5	Hospital Acquired Condition (HAC) assignment criteria #1 00 = Criteria to be assigned as an HAC not met 08 = Infection following CABG 10 = DVT/PE following knee or hip replacement 11 = Infection following bariatric surgery 12 = Infection following certain orthopedic procedures of spine, shoulder or elbow 13 = Surgical site infection (SSI) following cardiac implantable electronic device (CIED) procedures 14 = Iatrogenic pneumothorax w/ venous catheterization
6 and 7	Hospital Acquired Condition (HAC) assignment criteria #2 Refer to positions 4 and 5 for a list of values.

Position	Description (20 characters per procedure)
8 and 9	Hospital Acquired Condition (HAC) assignment criteria #3 Refer to positions 4 and 5 for a list of values.
10 and 11	Hospital Acquired Condition (HAC) assignment criteria #4 Refer to positions 4 and 5 for a list of values.
12 and 13	Hospital Acquired Condition (HAC) assignment criteria #5 Refer to positions 4 and 5 for a list of values.
14-20	Filler

## Ancillary buffer

The version number identifies the version of the grouper that is running.

**Table 7. Additional flag information**

Length in bytes	Description
5	1 byte reserved space (zero-filled) followed by 4-byte final DRG (after HAC logic applied)
1	Final DRG Medical/Surgical Indicator 0 = Error DRG (998 or 999) 1 = Medical DRG 2 = Surgical DRG
4	1 byte reserved space (zero-filled) followed by 3-byte initial DRG (prior to HAC logic)
5	1 byte reserved space (zero-filled) followed by 4-byte initial DRG (prior to HAC logic)
1	Initial DRG Medical/Surgical indicator 0 = Error DRG (998 or 999) 1 = Medical DRG 2 = Surgical DRG
8	Version ID returned by the grouper (PPPVVUU) PPP = 001 (MS-DRG) VVV = 370 (Grouper version 37.0) UU = 00 (update 00)



# Chapter 2: Installing the MS-DRG Software

Downloading and installing the Medicare Severity Diagnosis Related Groups (MS-DRG) Software (the grouper) consists of three steps:

1. Downloading and unzipping the file to your local machine
2. Allocating and FTPing the files to the mainframe
3. Link-editing the Assembler subroutines and testing the grouper

The first two steps are discussed in this chapter.

Step 3, the procedure for link-editing and testing the programs, depends on the way in which the grouper is implemented at your installation. Chapter 3 (page [39](#)) explains how to use the grouper utility and contains JCL for using it to run the test database. Chapter 4 shows how to link-edit the grouper programs for use as subroutines for a higher-level language. Two COBOL programs using the test database are included on the media, and the JCL for using them to test the installation is included in chapter 4.

The content of the downloaded file folder is shown in the following table.

**Table 8. MS-DRG media contents**

File	File name	LRECL	BLKSIZE	Description
1	OBJLIB	80	27920	Object library
2	SRCLIB	80	27920	Source library
3	LOADLIB	0	6233	Load library

The content of the miscellaneous folder is shown in the following table.

**Table 9. MS-DRG miscellaneous folder contents**

File	File name	LRECL	BLKIZE	Description
1	TESTDB	1760	26400	Test database
2	MDCDSC	80	27920	MDC titles
3	DRGDSC3	200	27800	DRG titles (3-digit)
4	DRGDSC4	200	27800	DRG titles (4-digit)
5	DRGLOGIC	200	27800	DRG logic
6	DXATTLST	100	27900	Diagnosis attribute list

File	File name	LRECL	BLKIZE	Description
7	DXLIST	20	27980	Diagnosis list
8	DXPATTERN	60	27960	Diagnosis pattern
9	EXCLUSN	15	27990	Exclusion
10	PRATTLST	120	27960	Procedure attribute list
11	PRCLSTRS	25	27975	Procedure clusters
12	PRLIST	20	27980	Procedure list
13	PRPATTERN	160	27840	Procedure pattern

## eDownload instructions

This section contains instructions for downloading program files from the Internet or from a CD for the Medicare Severity Diagnosis Related Groups (MS-DRG) Software.

### Grouper program installation

All required software for executing the MS-DRG 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
- Miscellaneous
  - Test database file
  - MS-DRG grouper tables
  - DRG and MDC descriptions

#### *Load library*

The load library is a sequential file, FTPLOAD.

The load library consists of the load modules for the MS-DRG Grouper. The entire load library is optional if you intend to use the object modules.

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(12,1),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 PDS on the mainframe using the following file characteristics:

- DSN = [e.g. YOURID.GROUPER.LOADLIB]
- RECFM = U
- BLKSIZE = 6233
- SPACE = (CYL(15,3,2),RLSE)

4. Create a BUILD PDS 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 PDS 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 shown below.

```
//JOB CARD FOR YOUR INSTALLATION
// *****
//* *** RECEIVE FTP'D SEQUENTIAL FILES TO CREATE LOAD LIBRARY PDS ***
// *****
//BDLOAD EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
RECEIVE INDATASET ('YOURID.GROUPER.FTPLOAD')
DATASET ('YOURID.GROUPER.LOADLIB')
/*
```

5. After you modify the BUILD PDS, execute the JCL.

**Table 10. Load library contents**

Number	Name	Description
1	ALTTEST	Sample COBOL program (alternate interface)
2	COBTEST	Sample COBOL (standard interface) program
3	DT370CA	Control program (alternate interface)
4	DT370CN	Control program (standard interface)

### Object library

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

**Table 11. Object library contents**

Program	Description
DT370CN	The main control program (standard interface)
DT370GR	The grouper program
DT370RT	The grouper tables
DT370CA	The main control program (alternate interface)
DT370UT	The grouper utility interface

The first three programs (DT370CN, DT370GR, DT370RT) comprise the main grouper executor using the standard interface. Substitute DT370CA for DT370CN (that is, use DT370CA, DT370GR, and DT370RT) to compile the main grouper executor using the alternate (re-entrant, macro-free) interface. DT370UT is a utility program that can serve as an interface if your input data meets specific criteria. Chapter 3 (page [39](#)) discusses this program.

All of the programs contained on the media were written in IBM Basic Assembler. There may be some reprogramming involved for those installations that do not have IBM equipment. The source code for each of the programs is provided.

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

The following steps download the object library.

1. Allocate a PDS on your mainframe with the following characteristics:
  - DSN = [e.g. YOURID.GROUPER.OBJLIB]

- RECFM = FB
  - LRECL = 80
  - BLKSIZE = 27920
  - SPACE = (CYL(10,1,2),RLSE)
2. FTP in **BINARY mode** all of the files in the object library folder into the PDS 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 seven members, as listed in the following table.

**Table 12. Source library contents**

Program	Description
DT370CN	The main control program (standard interface)
DT370GR	The grouper program
DT370RT	The grouper tables
DT370UT	The grouper utility interface program
DT370CA	The main control program (alternate interface)
COBTEST	The COBOL test interface program (standard interface)
ALTTEST	The COBOL test interface program (alternate interface)

Comments are also included in the source programs, DT370CN and DT370UT, describing the modifications needed to convert the programs to VSE.

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

1. Allocate a PDS on your mainframe with the following characteristics:
  - DSN = [e.g. YOURID.GROUPER.SRCLIB]
  - RECFM = FB

- LRECL = 80
  - BLKSIZE = 27920
  - SPACE = (CYL(48,1,4),RLSE)
2. FTP in ASCII mode all of the files in the source library folder into the PDS allocated in step 1.

## Miscellaneous files installation

### 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. It is also shown in the SYSUT2 DD card in JCL library member **DBLOAD**.
  - DSN=YOURID.GROUPER.TESTDB
  - RECFM=FB
  - LRECL=1760
  - BLKSIZE=26400
  - SPACE=(CYL,(65,1),RLSE)
2. FTP the TESTDB file in ASCII mode from the miscellaneous folder to the mainframe, YOURID.GROUPER.TESTDB.

The following table provides a record description for the test database.

**Table 13. Record layout for grouper test database**

Field	Location	Name	Description
1	1-3	AGE	Age on admission, in years
2	4-4	SEX	Gender
3	5-6	DSP	Discharge status (disposition)
4	7-7	POALOG	POA logic indicator
5	8-15	ADATE	Admission date (YYYYMMDD)
6	16-23	DDATE	Discharge date (YYYYMMDD)
7	24-223	DX1-25	Diagnosis codes (DX1=Principal)
32	224-398	PROC1-25	Procedure codes
57	399-598	PRDATES (1-25)	Procedure dates (YYYYMMDD)

Field	Location	Name	Description
82	599-600	RTC	Return code from the grouper
83	601-602	MDC	MDC number returned by the grouper
84	603-606	DRG	Final DRG number returned by the grouper
85	607-611	GRFLGS	Output grouper flags
86	612-1236	DXFLGS	Output diagnosis flags (25x25)
111	1237-1736	PRFLGS	Output procedure flags (25x20)
136	1737-1760	BUFF	Output ancillary buffer

**Note:** All diagnoses are left-justified in the first seven characters of an eight-character field, with each diagnosis' POA indicator occupying the eighth character of the field. Procedures are left-justified in seven-character fields. Unused characters in the diagnosis and procedure fields must be blank.

### MDC Description File

The following steps load the MDC description file to the mainframe.

1. Allocate a sequential file (PS) on your mainframe using the attributes below.
  - DSN=YOURID.GROUPER.MDCDSC
  - RECFM=FB
  - LRECL=80
  - BLKSIZE=27920
  - SPACE=(TRK,(1,2),RLSE)
2. FTP the MDCDSC file in ASCII mode from the miscellaneous folder to the mainframe YOURID.GROUPER.MDCDSC.

**Table 14. Record layout for MDCDSC**

Column	Description
1-2	MDC number
3-3	Comma (,)
4-80	MDC title

### DRG 3 Description File

The following steps load the DRGDSC3 description file to the mainframe.

1. Allocate a sequential file (PS) on your mainframe using the attributes below.
  - DSN=YOURID.GROUPER.**DRGDSC3**
  - RECFM=FB
  - LRECL=200
  - BLKSIZE=27800
  - SPACE=(TRK,(4,2),RLSE)
2. FTP the DRGDSC3 file in ASCII mode from the miscellaneous folder to the mainframe YOURID.GROUPER.**DRGDSC3**.

**Table 15. Record layout for DRGDSC3**

Column	Description
1-3	DRG number
4-4	Comma (,)
5-7	Constant 'MDC'
8-8	Blank
9-10	MDC number
11-11	'M' (medical) or 'P' (surgical)
12-12	Comma (,)
13-200	DRG title

### DRG 4 Description File

The following steps load the DRGDSC4 description file to the mainframe.

1. Allocate a sequential file (PS) on your mainframe using the attributes below.
  - DSN=YOURID.GROUPER.**DRGDSC4**
  - RECFM=FB
  - LRECL=200
  - BLKSIZE=27800
  - SPACE=(TRK,(4,2),RLSE)

2. FTP the DRGDSC4 file in ASCII mode from the miscellaneous folder to the mainframe YOURID.GROUPER.DRGDSC4.

**Table 16. Record layout for DRGDSC4**

Column	Description
1-4	DRG number
5-5	Comma (,)
6-200	DRG title

## ICD-10-CM/PCS MS-DRG Grouper Tables

Under ICD-9-CM, the MS-DRG Grouper Tables were included with the distribution of the mainframe MS-DRG grouper and were collectively referred to as the “EBCDIC Tables”. Though developed in ASCII on non-mainframe computers, the tables were necessarily converted to the EBCDIC character representation of mainframes in order to be included in the mainframe distribution.

Whether expressed in ASCII or EBCDIC, the ICD-10-CM/PCS MS-DRG Grouper Tables consist of nine files, as documented below. One of these, drglogic, specifies, via a non-platform-specific pseudo-language, the decisions the grouper makes to determine the MS-DRG from attributes of the patient and the codes on the patient record. All of the others are comma-delimited tables relating codes to attributes.

In the documentation of the comma-delimited files below, the information between successive commas is called a field. Field 1 is the information before the first comma, field 2 is that between the first and second comma, and so on. Information itself containing a comma will be enclosed in double quotation marks. No fields so enclosed are allowed themselves to contain double quotation marks.

### Attributes

Since there are approximately 140,000 ICD-10-CM/PCS codes, with new ones being added every year, yet only a few codes on any given patient record, efficiency dictates that we transform the codes on the record into a small set of attributes, and then base the DRG-assignment logic on those attributes. An attribute is a patient-level yes/no result – either the patient record has the attribute or it does not. Hence in most implementations of the grouper, the attributes are represented with individual bits.

The grouper maintains five sets of attributes:

- Patient characteristics based on age, sex and discharge status, computed once when the patient data is provided.

- Principal diagnosis attributes, obtained by looking up the first listed diagnosis and saving its attributes from the diagnosis tables.
- Secondary diagnosis attributes, obtained by looking up each diagnosis on the record after the first, and combining the attributes found in the diagnosis tables using a Boolean inclusive-OR.
- “Any” diagnosis attributes, obtained via a Boolean inclusive-OR of the principal diagnosis attributes and the secondary diagnosis attributes.
- Procedure attributes, obtained by looking up each procedure on the record, and combining the attributes found in the procedure tables using a Boolean inclusive-OR for non-restricted procedures. Procedure clusters (see below) must also be discovered and their attributes included. This must be done each time the working MDC changes since procedure restriction is MDC dependent.

### *Procedure clusters*

When found on a patient record, some collections of ICD-10-PCS procedure codes have a different set of attributes, independent of those of the codes that make them up (their “components”). These are called procedure clusters. A routine in the grouper, upstream of the DRG assignment logic, searches the patient record for clusters. (Multiple clusters, and duplicates of the same cluster, are possible.) When a cluster is found, it is added to the list of procedures found on the record. Clusters may be “restricted” by MDC. A restricted cluster inhibits the use of its component attributes for the MDC’s DRG assignment logic.

For example, cluster 0SRT0JZ+0SPC0JZ may be recognized on the patient record if both codes appear (in any order and not necessarily together). This creates a new “procedure code” @0045. The cluster @0045 has a different set of attributes than either 0SRT0JZ or 0SPC0JZ, and is further “restricted” for MDC 08. If the grouper logic determines that the MDC is 08, it ignores the attributes of 0SRT0JZ and 0SPC0JZ and only uses those of @0045. These take the grouper to DRGs 466-468 rather than DRG 463-465. If the PDX were not for MDC 08, however, the cluster would not restrict the individual interpretation of the component codes and their own attributes could come into play as well as those of @0045.

### *Grouper logic (drglogic)*

The decisions that the grouper makes to assign a patient discharge to a DRG are specified in the file *drglogic*. The specifications resemble a computer language like C, C# or Java, but with a few special conventions to keep the file small:

- Two slashes (//) introduce a comment, which continues to the end of the line.
- A key at the beginning of the file explains the attribute notation.
- Logic introduced by “{” continues until the matching “}” is reached.
- A statement of the form “return MDC|DRG|BASEDRG=3|153|152” means to assign the values on the right of the equal sign respectively to the variables on the left, then return to

the main loop. In this example, MDC would be set to 3, DRG to 153 and BASEDRG to 152 before returning.

### Diagnosis attributes (*dxattlst*)

Each row of the table identifies a diagnosis attribute potentially used by the grouper logic.

**Table 17. Diagnosis attributes**

Field	Contents
1	Attribute number. Bits in mainframe implementation are stored left-to-right in this order.
2	Attribute name as used in <i>drglogic</i>
3	Attribute description.

### Diagnosis table (*dxlist*)

Each valid ICD-10-CM diagnosis code is found in the table.

**Table 18. Diagnosis table**

Field	Contents
1	Diagnosis code. ICD-10-CM codes are represented without their decimal point.
2	0 = fields 3 and 4 relate to both sexes. 1 = fields 3 and 4 relate to male patients only. 2 = fields 3 and 4 relate to female patients only.
3	The "exclusion category" to be used to find CC/MCC exclusions when this diagnosis is used as a PDX (principal diagnosis). If 0, then no CC or MCC is excluded. For other values, see <i>exclusn</i> .
4	The pattern number of the row in <i>diagnosis_patterns.txt</i> which gives the attributes of this diagnosis

### Diagnosis pattern table (*dxpatern*)

Each distinct combination of MDC, diagnosis category and attributes has its own row. The pattern number is arbitrary and is used only as a link between *dxlist* and this table.

**Table 19. Diagnosis pattern table**

Field	Contents
1	Pattern number
2	MDC
3	Diagnosis category (DXCAT in <i>drglogic</i> ).
4	HAC group. 0 if diagnosis is not in a HAC group
5	List of attributes, separated by the vertical bar (   )

*Procedure attributes (prattlst)*

Each row of the table identifies a procedure attribute potentially used by the grouper logic.

**Table 20. Procedure attributes**

Field	Contents
1	Attribute number. Bits in mainframe implementation are stored left-to-right in this order.
2	Attribute name as used in <i>drglogic</i> .
3	Attribute description.

*Procedure table (prlist)*

Each valid ICD-10-PCS procedure code occupies one row in the table.

**Table 21. Procedure table**

Field	Contents
1	Procedure code or cluster. Procedure clusters are identified by an "@" sign, followed by a 4-digit number. This is a "cluster ID", which will match the cluster ID field in <i>prclstrs</i> .
2	The pattern number of the row in <i>prpattern</i> which gives the attributes of this procedure or procedure cluster.

*Procedure pattern table (prpattern)*

Each distinct combination of procedure attributes has its own row. The pattern number is arbitrary and is used only as a link between *prlist* and this table.

**Table 22. Procedure pattern table**

Field	Contents
1	Pattern number
2	List of procedure attributes, separated by the vertical bar (   ). The special attribute "incluster" indicates that the procedure is in at least one cluster.

*Procedure cluster definitions (prclstrs)*

Each component of each distinct procedure cluster will have a row in this table.

**Table 23. Procedure cluster definitions**

Field	Contents
1	Procedure code.
2	Cluster ID of one cluster the procedure is a component of.
3	The choices group for the cluster this procedure is a member of. For example, "1" means that this procedure is one of the first set of procedures listed for the cluster, a "2" means it is one of the second set of procedures listed for the cluster. A cluster may have only one procedure in a choices set. The procedures in the cluster (its components) may be recognized on the patient record in any order and need not be listed together.
4	The number of procedures required to recognize the cluster. For example, a "3" means that the cluster is recognized if there is at least one procedure on the record from its first set, one from its second and one from its third.
5	The list of MDCs for which the cluster is restricted, separated by the vertical bar (   ). "ALL" signifies that the cluster is restricted for all MDCs. "F" signifies that the cluster is listed in Appendix F of the MS-DRG Definitions Manual.

*CC/MCC exclusions (exclusn)*

A principal diagnosis (PDX) may exclude certain secondary diagnoses (SDX) from being considered complications/co-morbidities (CC) and/or major complications/co-morbidities (MCC). The diagnosis table in *dxlist* gives an “exclusion category” for each PDX. A zero indicates that the diagnosis has no CC/MCC exclusions. All numbers 2 and over refer to the first field of this table. All SDX listed in this table with a given exclusion category are prevented from being considered CCs or MCCs when the PDX has the exclusion category listed.

**Table 24. CC/MCC exclusions**

Field	Contents
1	Exclusion category.
2	Excluded secondary diagnosis. ICD-10-CM codes are listed without their decimal point.

**Uploading grouper tables/files to the mainframe***DRG logic file*

The following steps load the DRG logic file to the mainframe.

1. Allocate a sequential dataset using the following attributes:
  - DSN=YOURID.GROUPER.**DRGLOGIC**
  - LRECL=200
  - BLKSIZE=27800
  - RECFM=FB
  - SPACE=(CYL(1),RLSE)
2. FTP the DRGLOGIC file from the miscellaneous folder in ASCII mode into a mainframe sequential dataset, “YOURID.GROUPER.**DRGLOGIC**”.

*Diagnosis attribute list*

The following steps load the Diagnoses attribute list to the mainframe.

1. Allocate a sequential dataset using the following attributes:
  - DSN=YOURID.GROUPER.**DXATTLST**

- LRECL=100
  - BLKSIZE=27900
  - RECFM=FB
  - SPACE=(TRK(1),RLSE)
2. FTP the DXATTLIST file from the miscellaneous folder in ASCII mode into a mainframe sequential dataset, "YOURID.GROUPER.DXATTLST".

### *Diagnosis list file*

The following steps load the Diagnosis list file to the mainframe.

1. Allocate a sequential dataset using the following attributes:
  - DSN=YOURID.GROUPER.DXLIST
  - LRECL=20
  - BLKSIZE=27980
  - RECFM=FB
  - SPACE=(CYL(2),RLSE)
2. FTP the DXLIST file from the miscellaneous folder in ASCII mode into a mainframe sequential dataset, "YOURID.GROUPER.DXLIST".

### *Diagnosis pattern list*

The following steps load the Diagnosis pattern list to the mainframe.

1. Allocate a sequential dataset using the following attributes:
  - DSN=YOURID.GROUPER.DXPATTERN
  - LRECL=60
  - BLKSIZE=27960
  - RECFM=FB
  - SPACE=(TRK(1),RLSE)
2. FTP the DXPATTERN file from the miscellaneous folder in ASCII mode into a mainframe sequential dataset, "YOURID.GROUPER.DXPATTERN".

### *CC/MCC exclusions file*

The following steps load the CC/MCC exclusions file to the mainframe.

1. Allocate a sequential dataset using the following attributes:

- DSN=YOURID.GROUPER.**EXCLUSN**
  - LRECL=15
  - BLKSIZE=27990
  - RECFM=FB
  - SPACE=(TRK(50),RLSE)
2. FTP the EXCLUSN file from the miscellaneous folder in ASCII mode into a mainframe sequential dataset, "YOURID.GROUPER.**EXCLUSN**".

### *Procedure attribute list*

The following steps load the Procedure attribute list to the mainframe.

1. Allocate a sequential dataset using the following attributes:
  - DSN=YOURID.GROUPER.**PRATTLST**
  - LRECL=120
  - BLKSIZE=27960
  - RECFM=FB
  - SPACE=(TRK(1),RLSE)
2. FTP the PRATTLIST file from the miscellaneous folder in ASCII mode into a mainframe sequential dataset, "YOURID.GROUPER.**PRATTLST**".

### *Procedure list file*

The following steps load the Procedure list file to the mainframe.

1. Allocate a sequential dataset using the following attributes:
  - DSN=YOURID.GROUPER.**PRLIST**
  - LRECL=20
  - BLKSIZE=27980
  - RECFM=FB
  - SPACE=(CYL(2),RLSE)
2. FTP the PRLIST file from the miscellaneous folder in ASCII mode into a mainframe sequential dataset, "YOURID.GROUPER.**PRLIST**".

### *Procedure pattern list*

The following steps load the Procedure pattern list to the mainframe.

1. Allocate a sequential dataset using the following attributes:
  - DSN=YOURID.GROUPER.**PRPATTERN**
  - LRECL=160
  - BLKSIZE=27840
  - RECFM=FB
  - SPACE=(TRK(5),RLSE)
2. FTP the PRPATTERN file from the miscellaneous folder in ASCII mode into a mainframe sequential dataset, "YOURID.GROUPER.**PRPATTERN**".

### *Procedure clusters file*

The following steps load the Procedure clusters file to the mainframe.

1. Allocate a sequential dataset using the following attributes:
  - DSN=YOURID.GROUPER.**PRCLSTRS**
  - LRECL=25
  - BLKSIZE=27975
  - RECFM=FB
  - SPACE=(TRK(1),RLSE)
2. FTP the PRCLSTRS file from the miscellaneous folder in ASCII mode into a mainframe sequential dataset, "YOURID.GROUPER.**PRCLSTRS**".



# Chapter 3: Using and testing the grouper utility

Installations with data that conforms to the grouper requirements provided in chapter 1 (page [9](#)) and whose output record length does not exceed 2992 bytes, may implement the grouper as a utility program that receives all information pertaining to the input record layout from the job's SYSIN stream. To use the grouper utility, you must have copied file 1 (the grouper object library) from the media to disk (page [21](#)).

## Link-editing the grouper utility

The sample JCL for creating a load module for the grouper utility is shown in the following figure.

```
//JOB CARD FOR YOUR INSTALLATION
//* *****
//* THIS JOB CREATES A GROUPER UTILITY LOAD MODULE   *
//* *****
//LKED EXEC PGM=HEWL, PARM=' LIST, MAP, AMODE=31, RMODE=ANY' ,
// REGION=1024K
//SYSLMOD DD DSN=GROUPER.UTIL.LOAD, DISP=OLD
//SYSUT1 DD UNIT=DISK, SPACE=(CYL, (1, 1))
//SYSPRINT DD SYSOUT=*
//OBJECT DD DSN=GROUPER.OBJLIB, DISP=OLD
//SYSLIN DD *
INCLUDE OBJECT(DT37OUT, DT370CN, DT370GR, DT370RT)
ENTRY DT37OUT
NAME DT37OUT
/*
```

## Using the grouper utility

As previously mentioned, the grouper utility receives all information pertaining to the input and output record layouts from the job's SYSIN stream. When using the grouper utility, you must provide 16 SYSIN control statements shown in the following table. These statements must be present in the order shown. Each control statement consists of a 3-character keyword followed by at least one 4-digit field, right-justified and zero-filled, indicating the starting position of the variable.

**Table 25. Control statements required by the grouper utility**

Control statement	Keyword	Identifies the starting position(s) of...
1	DDX	Each 8-byte diagnosis code
2	SRG	Each 7-byte procedure code
3	AGE	The age field
4	SEX	The sex field
5	DSP	The discharge status field
6	POA	Present on admission logic
7	ADT	Admission date
8	DDT	Discharge date
9	SDT	Procedure dates
10	RTC	The grouper return code
11	MDC	The MDC number returned by the grouper
12	DRG	The DRG number returned by the grouper
13	GFL	Grouper flags
14	DFL	Diagnosis flags
15	SFL	Procedure flags
16	BUF	Grouper buffer

## Control statement examples

The following examples of the control statements use the 1760-byte record from the test database as input. The first 598 bytes contain the data that must be passed to the grouper, and the next 1162 bytes contain the information filled in by the previous grouper. The output record is 1162 bytes larger, with those 1162 bytes containing the data returned by the new grouper when you run the test.

### *The discharge diagnosis control statement (DDX)*

The DDX control statement specifies the starting position of each discharge diagnosis code in the patient record to be used in the grouping process. Blanks must be inserted between each position specified. The grouper assumes that the first specified diagnosis is the principal discharge diagnosis. You may specify up to 24 secondary diagnoses to be considered in the

grouping process so there may be at most 25 diagnosis positions specified on the control statement. For example, the DDX control statement shown below indicates that the principal diagnosis started at position 24 and that there were 24 secondary diagnoses to be used by the grouper, which began at position 32.

The grouper assumes that each diagnosis code specified is left-justified in a **8-byte field**. All codes must be blank-filled. Zero-filled codes are not allowed. The 8th byte in each field is the POA indicator.

```
Contents  DDX 0024 0032 0040 0048 0056 0064 0072 0080 0088 0096 0104 0112 0120 0128 0136 *
          DDX 0144 0152 0160 0168 0176 0184 0192 0200 0208 0216
```

When there are more than 15 diagnoses, the asterisk (\*) must be placed in column 80.

### *The procedure control statement (SRG)*

The SRG control statement specifies the starting position of each procedure code in the patient record to be used in the grouping process. As with the diagnosis control statement, you specify each starting position as a 4-digit number. Blanks must be inserted between each position specified. You may provide up to 25 procedures for use by the grouper. For example, the SRG control statement shown below indicates that there were 25 procedure codes to be used in the grouping process, with the first procedure beginning at position 224, the second procedure beginning at position 231, and so on.

The grouper assumes that each procedure code specified is left-justified in a **7-byte field**. Short codes must be blank-filled. Zero-filled codes are not allowed.

```
Contents  SRG 0224 0231 0238 0245 0252 0259 0266 0273 0280 0287 0294 0301 0308 0315 0322 *
          SRG 0329 0336 0343 0350 0357 0364 0371 0378 0385 0392
```

When there are more than 15 procedures, the asterisk (\*) must be placed in column 80.

### *The age control statement (AGE)*

The AGE control statement specifies the starting position of the field containing the patient age. Only ages between 0 and 124 are considered valid for grouping. The age field is assumed to be three bytes in length, containing right-justified numerics, and may be either zero- or blank-filled. For example, the AGE control statement displayed below indicates that the 3-byte age field appears on the patient record starting at position 1.

```
Column      123456789
Contents     AGE 0001
```

### *The sex control statement (SEX)*

The SEX control statement specifies the starting position of the field containing the patient's sex. The grouper assumes that the sex field is one byte in length, containing the values 0 through 2 (unknown/male/female respectively). The test database SEX control statement is:

Column	123456789
Contents	SEX 0004

### *The discharge status control statement (DSP)*

The DSP control statement specifies the position of the discharge status on the patient's record. The grouper assumes this is a 2-byte, right-justified field, with values as specified in the "Required data formats" (page 9) table. Short codes (i.e., codes with fewer than two digits) may be either blank- or zero-filled. The test database DSP control statement is:

Column	123456789
Contents	DSP 0005

### *The present on admission control statement (POA)*

The POA control statement specifies the starting position of the field containing the Present on Admission logic flag. The grouper assumes the POA flag is one byte in length, containing the values specified in the "Required data formats" (page 9) table. The test database control statement is:

Column	1234567890
Contents	POA 0007

### *The admission date control statement (ADT)*

The ADT control statement specifies the starting position of the field containing the patient's admission date. The grouper assumes the admission date is 8 bytes in length, formatted as YYYYMMDD. The test database control statement is:

Column	1234567890
Contents	ADT 0008

### *The discharge date control statement (DDT)*

The DDT control statement specifies the starting position of the field containing the patient's discharge date. The grouper assumes the discharge date is 8 bytes in length, formatted as YYYYMMDD. The test database control statement is:

Column	1234567890
Contents	DDT 0016

### *The procedure dates control statement (SDT)*

The SDT control statement specifies the starting position of a 200-byte buffer containing the date of each procedure coded on the patient record. The grouper assumes each procedure date is 8 bytes in length, formatted as YYYYMMDD. The test database control statement is:

Column	1234567890
Contents	SDT 0399

## **Grouper output control statements**

It is important to note that none of the data returned by the grouper needs be written to the output record, although presumably you would want at least DRG and MDC numbers and the grouper return code (RTC). Regardless of whether you choose to output the data or not, a control statement with an output position must be supplied for each of the elements specified below (RTC, MDC, DRG, GFL, DFL, SFL, BUF).

You must ensure that the storage for all fields returned by the grouper can be contained on the output record. The utility program determines the output record length from the JCL DCB specifications for the output dataset. If the position specified is beyond the end of the output record but within the maximum record length allowed, the field is dropped when the output record is written.

### *The return code control statement (RTC)*

The RTC control statement specifies the location of a 2-byte field, which is used to store the grouper return code. The test database return code control statement is:

Column	123456789
Contents	RTC 0599

### *The MDC control statement (MDC)*

The MDC control statement specifies the starting position for the storage of the MDC number returned by the grouper. The MDC number is a 2-byte, right-justified numeric value. The test database MDC control statement is:

Column	123456789
Contents	MDC 0601

### *The DRG control statement (DRG)*

The DRG control statement specifies where on the output record the grouper should store the Final DRG number. The Final DRG number returned by the grouper is a 4-byte, right-justified numeric value. The test database DRG control statement is:

Column	123456789
Contents	DRG 0603

### *The grouper flags control statement (GFL)*

The GFL control statement specifies the starting position of the grouper flags. The grouper assumes this field to be 5 bytes in length. The test database control statement is:

Column	1234567890
Contents	GFL 0607

### *The diagnosis flags control statement (DFL)*

The DFL control statement specifies the starting position of the diagnosis flags. The grouper assumes this field to be 625 bytes in length. There are 25 diagnosis flags for each diagnosis on the record, up to a total of 25 diagnosis codes. The test database control statement is:

Column	1234567890
Contents	DFL 0612

### *The procedure flags control statement (SFL)*

The SFL control statement specifies the starting position of the procedure flags. The grouper assumes this field to be 500 bytes in length. There are 20 procedure flags for each procedure on the record, up to a total of 25 procedure codes. The test database control statement is:

Column	1234567890
Contents	SFL 1237

### The buffer control statement (BUF)

The BUF control statement specifies the starting position of the buffer of additional DRG information. The grouper assumes this field to be 24 bytes in length. The test database control statement is:

```
Column      1234567890
Contents    BUF 1737
```

## Running the grouper utility program

The following table shows the ABENDs (abnormal end of jobs) possible from the grouper utility program.

**Table 26. ABEND codes**

Code	Description
80A	Insufficient region size
001	Control statements missing or out of order
002	Non numeric data in position field on control statement
003	Missing control statement
004	Unsuccessful open of input database
005	Unsuccessful open of output database
006	Continuation character (*) found with less than 15 codes

The following sample JCL is for executing the grouper utility program.

```
//GO EXEC PGM=DT37OUT
//STEPLIB DD DSN=GROUPER.UTIL.LOAD,DISP=SHR
//IN DD DSN=YOURID.GROUPER.TESTDB,DISP=SHR
//OUT DD DSN=GROUPER.OUTTEST.DATA,
// DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
// DCB=(LRECL=1760,BLKSIZE=26400,RECFM=FB),
// SPACE=(CYL,(10,1),RLSE)
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FA,BLKSIZE=133,BUFNO=1)
//SYSIN DD *
DDX 0024 0032 0040 0048 0056 0064 0072 0080 0088 0096 0104 0112 0120 0128 0136 *
DDX 0144 0152 0160 0168 0176 0184 0192 0200 0208 0216
SRG 0224 0231 0238 0245 0252 0259 0266 0273 0280 0287 0294 0301 0308 0315 0322 *
SRG 0329 0336 0343 0350 0357 0364 0371 0378 0385 0392
AGE 0001
SEX 0004
DSP 0005
POA 0007
ADT 0008
DDT 0016
SDT 0399
RTC 0599
MDC 0601
DRG 0603
GFL 0607
DFL 0612
SFL 1237
BUF 1737
```

**Note:** The SYSIN control statements must not contain line numbers, as the entire 80 bytes is considered input. Failure to do this causes User ABEND 001.

# Chapter 4: Using the grouper with higher-level languages

The grouper executor may be implemented as a subroutine to be called from Assembler or a higher-level language program. This chapter shows how this may be done for a COBOL programming environment. To create the subroutines, you must have copied file 1 (the grouper objlib) from the media to disk (page [21](#)).

## General strategy for COBOL driving program

A typical COBOL grouping utility might operate as follows:

- Opens the input and output datasets
- Reads records from the input dataset
- Reformats and recodes the input data to a form acceptable to the grouper
- Calls the grouper
- Stores the grouper return information on the output record
- Writes a new dataset containing the original data and the grouping information

A COBOL program (COBTEST) using the sample database is included on the media.

The following sample JCL is for grouping test database in the COBOL environment.

```

//JOB CARD FOR YOUR INSTALLATION
//* *****
//* SAMPLE JCL FOR GROUPING TEST DATABASE IN THE COBOL      *
//* ENVIRONMENT.                                           *
//*                                                         *
//* BOTH OBJECT AND LOAD MODULES ARE TEMPORARY.          *
//* *****
//COBUCLG PROC
//* COBOL FOR MVS COMPILE AND LE370 LINK
//COB   EXEC   PGM=IGYCRCTL,PARM='RENT,NODYNAM'
//STEPLIB DD   DSN=IGYV3R4.SIGYCOMP,DISP=SHR
//SYSLIB  DD   DSN=GROUPER.SRCLIB,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSIN   DD   DSN=GROUPER.SRCLIB(COBTST),DISP=SHR
//SYSUT1  DD   UNIT=DISK,SPACE=(CYL,(1,1))
//SYSUT2  DD   UNIT=DISK,SPACE=(CYL,(1,1))
//SYSUT3  DD   UNIT=DISK,SPACE=(CYL,(1,1))
//SYSUT4  DD   UNIT=DISK,SPACE=(CYL,(1,1))
//SYSUT5  DD   UNIT=DISK,SPACE=(CYL,(1,1))
//SYSUT6  DD   UNIT=DISK,SPACE=(CYL,(1,1))
//SYSUT7  DD   UNIT=DISK,SPACE=(CYL,(1,1))
//SYSLIN  DD   DSN=&&LOADSET,UNIT=DISK,DISP=(MOD,PASS),
//   SPACE=(TRK,(3,3)),DCB=BLKSIZE=800
//*
//LKED   EXEC   PGM=IEWL,PARM='LIST,MAP,AMODE=31,RMODE=ANY',
//   COND=(5,LT,COB)
//SYSLIB  DD   DSN=CEE.SCEELKED,DISP=SHR
//SYSLMOD DD   DSN=&&GOSET(GO),DISP=(,PASS),UNIT=DISK,
//   SPACE=(CYL,(5,1,5))
//SYSUT1  DD   UNIT=DISK,SPACE=(CYL,(1,1))
//SYSPRINT DD  SYSOUT=*
//SYSLIN  DD   DSN=&&LOADSET,DISP=(OLD,DELETE)
//   DD   DDNAME=SYSIN
//OBJECT  DD   DSN=GROUPER.OBJLIB,DISP=OLD
//*
//GO EXEC   PGM=COBTST,COND=((5,LT,COB),(5,LT,LKED))
//STEPLIB DD   DISP=SHR,DSN=CEE.SCEERUN
//   DD   DISP=SHR,DSN=&&GOSET
//SYSPRINT DD  SYSOUT=*
//INFILE  DD   DSN=YOURID.GROUPER.TESTDB,DISP=SHR
//   PEND
//*
//PROG1   EXEC   COBUCLG,PROD=DRG370.I10
//LKED.SYSIN DD *
//   INCLUDE OBJECT(DT370CN,DT370GR,DT370RT)
//   ENTRY COBTST
//   NAME COBTST
//*

```

## Input to the grouper subroutines

The grouper control program (DT370CN) assumes that general purpose register 1 is pointing to a list of addresses with the structure shown in the following table.

**Table 27. MS-DRG software address list**

Offset	Fullword pointer to...
0	The buffer containing the ICD-10-CM diagnosis codes for the record to be grouped. The first code is assumed to be principal diagnosis.
4	4-byte binary (PIC 9(8) COMP) field indicating the number of diagnoses contained in the buffer discussed above. This can be the actual number of codes in the buffer, or the maximum number of codes that the buffer can hold. This number cannot be less than 1 nor greater than 25. If greater than 25, the software uses only the first 25 fields in the buffer and ignores the rest.
8	The buffer containing the procedure codes for the record to be grouped.
12	4-byte binary (PIC 9(8) COMP) field indicating the number of procedures present. This field has the same rules as for diagnoses, except that it may be zero.
16	3-byte numeric field containing the patient's age in years.
20	1-byte numeric field containing the patient's sex.
24	2-byte numeric field containing the patient's discharge status.
28	1-byte field containing the POA logic indicator
32	8-byte numeric field containing the patient's admission date (YYYYMMDD)
36	8-byte numeric containing the patient's discharge date (YYYYMMDD)
40	200-byte buffer containing the dates of the procedure codes. The buffer can hold up to a maximum of 25 dates, 8-bytes each (YYYYMMDD).
44	2-byte numeric field to hold the grouper return code.
48	2-byte numeric field to hold the MDC number.
52	4-byte numeric field to hold the DRG number.
56	5-byte field to hold the grouper flags.
60	625-byte field to hold the diagnosis flags.
64	500-byte field to hold the procedure flags.
68	24-byte field to hold the buffer of additional DRG information.

**Note:** COBOL applications programmers need not concern themselves with implementing this structure since COBOL automatically creates it when a CALL USING statement is issued.

You must ensure that each diagnosis code is left-justified in a 8-byte field and that all of the diagnoses are in contiguous locations in the buffer whose address is in the first pointer described above. Empty fields may be interspersed throughout the buffer. A detailed discussion of the way in which fields in the buffer are processed is located at the end of this chapter.

Similarly, each procedure code must be left-justified in a 7-byte field, and all of the procedure codes must be in contiguous locations in the buffer whose address is in the third pointer described above.

Each diagnosis and procedure code must be blank-filled if it is shorter than the maximum field length. *Zero filling is not allowed.*

The patient's age must be right-justified in a 3-byte field. Valid ages for grouping are between 0 and 124. The age may be either zero- or blank-filled.

The patient's sex must be contained in a 1-byte field, in the range 0 through 2 (Unknown/Male/Female, respectively).

The discharge status must be contained in a 2-byte field which is coded according to the conventions shown in the "Required data formats" (page [9](#)) table. The code must be right-justified and may be either zero- or blank-filled.

## Output from the grouper subroutines

On return from the grouper executor, the DRG, MDC, return code, and the grouper, diagnosis, and procedure flags fields are filled in, along with the buffer of additional DRG information. The DRG and MDC numbers are right-justified. The grouper return code is filled in according to the conventions detailed in chapter 1 (page [9](#)).

## Using the alternate interface

The alternate grouper control program, (DT370CA) operates the same as the standard grouper control program (DT370CN) except that it does not contain any macros and is written to be re-entrant, so it should run in a wider variety of mainframe environments. Whereas the standard interface uses GETMAIN to obtain a 24,000 byte work area, the alternate interface requires that the calling program provide the work area. It must do so by providing two additional addresses in the list pointed to by general register 1. For details see the "MS-DRG software address list" (page [48](#)) table.

The following table gives the additional work area parameters required by the alternate interface.

**Table 28. Work area parameters**

Offset	Full word pointer to...
72	A buffer of at least 24,000 bytes.
76	4-byte binary (PIC 9(8) comp) field containing the actual length in bytes of the work area. The value of this field should not be less than 24,000 bytes, though larger values are acceptable.

To use the alternate interface, substitute DT370CA for DT370CN and provide these two extra parameters. See the COBOL program ALTTEST, provided in the source library, for an example of how to set up a work area and pass it to DT370CA.

Assembler programmers should note that the length of the work area is not given in the full word at the offset 76 from R1 but rather a pointer to the full word containing the length is given at offset 76.

Sample JCL for running ALTTEST may be created by modifying the JCL for grouping the test database in the COBOL environment (page 47). To modify the JCL, change all occurrences of COBTEST to ALTTEST and change DT370CN to DT370CA.

## Executor processing of the diagnosis and procedure buffers

The way in which the grouper retrieves diagnosis and procedure codes for processing is to loop through the related buffers using the counts addressed by the second and fourth pointers. If any diagnosis or procedure field is all zeroes or all blanks, then that field is considered empty and the code is flagged as invalid and is ignored. Codes are saved in an internal work area that is subsequently used for construction of the record mask (page 53). Because processing is done this way, it is possible to pass a buffer that contains both valid and empty fields.

For example, assume there is a record containing a maximum of five diagnosis codes, three of which are coded for this abstract. The number of diagnoses passed would be five, and the buffer could look like any of the following:

```
3310  Y40210  Y5601  N
3310  Y      40210  Y5601  N
3310  Y00000 40210  Y5601  N00000
```

The principal diagnoses must be in the first field of the buffer. If the field is empty or invalid, the record is assigned DRG 999 (ungroupable) with a return code of 7 (invalid principal diagnosis).



# Chapter 5: The MS-DRG grouper executor

To use the information in this chapter, you should have:

- A working knowledge of IBM Basic Assembler Language
- At least a rudimentary understanding of the underlying logic on which all DRG decisions are based
- Access to the *Medicare Severity Diagnosis Related Groups Definitions Manual*, which explains the principles on which all decisions are made

The executor essentially makes its decisions by comparing indicators for each DRG within an MDC. Indicators are set by the elements found on the patient record. These sets of indicators are referred to as masks. The content of the masks is listed in the MS-DRG grouper tables (page [29](#)).

The tables are represented as hexadecimal constants in the module DT370RT and are present in memory when the grouper is loaded for execution. All table lookups are in-memory binary searches.

The executor begins its basic task by creating masks that are indicative of the conditions found on the patient record. These are called the record masks.

Once the record masks have been constructed, the corresponding DRG masks for the MDC indicated by the principal diagnosis are compared to them, until a match is found or the DRG masks for the MDC are exhausted.

Because the internal format of the grouper tables is optimized in DT370RT for fast lookups and is therefore difficult to read, the nine principal tables included in DT370RT are provided as flat files on the media. See chapter 2 (page [21](#)) for table layout details.

## Construction of the record mask

The following list describes how the executor constructs the record masks.

1. Sex is tested for validity (0-2).
  - An error indicator is turned on if sex is out of that range.
  - If not, the appropriate indicator is set in the record mask.
2. Discharge status is tested for validity (01-07, 20, 21, 30, 43, 50, 51, 61-66, 69, 70, 81-95)
  - An error indicator is turned on if discharge status is out of range.
  - Otherwise, the appropriate indicators are set in the record mask.
3. The first listed diagnosis (assumed principal) is looked up in the Diagnosis Table.

- If no entry is found, the record is assigned DRG 999, RTC 7 and no further processing occurs.
  - If an entry is found, but the MDC number is 0, the record is assigned DRG 999, RTC 7 and no further processing occurs.
  - Otherwise, the MDC and DXCAT are saved and the indicators for this diagnosis code are moved to the mask where principal diagnosis indicators are positioned.
4. All secondary diagnoses are looked up in the Diagnosis Table and their bit indicators “OR’d” together in the mask reserved for secondary diagnosis indicators. Additionally, if any of the secondaries is a complication or comorbidity, the CC exclusion subroutine is called to determine if the CC flag in the record mask should be set. A complete discussion of the CC exclusion subroutine appears later in this chapter (page [55](#)).
- Any secondary diagnosis for which there is no Diagnosis Table entry does not cause an error, but is instead ignored. MDC and DXCAT numbers are of no importance for secondaries.
5. Once all diagnoses have been processed, the indicators for principal and secondary are “OR’d” together in yet another indicator section mask for ALLDX criteria.
6. All procedure codes are looked up in Procedure Table and their bit indicators “OR’d” together in the mask reserved for procedure indicators. As with secondary diagnoses, invalid procedure codes do not generate errors, but are ignored.

## DRG determination

Once the record masks have been constructed, the executor loops through the DRG masks for the MDC indicated by the principal diagnosis, comparing them with the record masks.

1. The comparison is done by moving the record mask to a work area and ANDing it with the current DRG mask.
2. The result of the ANDed work mask is then compared with the DRG mask.
  - If the results are identical, the associated DRG number is assigned and the processing to find and return the diagnosis and procedure flags is executed.
  - Otherwise, looping continues until a match is found or the DRG list is exhausted, at which time DRG 999 is assigned.

The rest of this section discusses some special conditions in the grouper logic.

### Testing for the ONLY surgery condition

When the DRG mask indicates that ONLY specific surgeries can be present, the executor loops through the saved O.R. surgeries from the record, making decisions as follows:

1. The O.R. portion of the DRG mask is moved to a work area.

2. The work mask is ANDed with the mask of the saved O.R. surgery.
  - If the result of the ANDing is zero, this indicates that the surgery found on the record is other than the ONLY surgery allowed. The executor ceases looping and gets the next DRG mask.
  - Otherwise, the process continues until all saved O.R. surgeries have been tested.

## Testing for the ONLY DX condition

The testing for this condition is virtually identical to that done for the ONLY surgery condition, except that the comparison is done on saved diagnoses against the ALLDX portion of the DRG mask.

## Testing for the OWISE condition

This condition exists for DRGs 794, 963-965 and 997. This is essentially the “fall through” DRG for the MDC and is assigned when no other DRG criteria have been met. The “anydx” bit in the DRG mask is turned on, leaving a mask with only that bit on, thereby guaranteeing a match.

## Testing for the ANYCOMB condition

This condition exists only for DRG 461-462 in MDC 8. The test is done by comparing all coded O.R. procedures with the procedure portion of the DRG mask and adding one to an accumulator for each procedure that has a matching mask. If the resulting count is less than two, this record does not meet the "anycomb" condition, and the next DRG mask is retrieved.

## CC exclusion subroutine

A large subset of the diagnosis codes are flagged as complication/comorbidity codes (CC) or major complication/comorbidity codes (MCC). Many of these codes are not really CC/MCC codes at all times because there are many conditions for which the secondary diagnosis is a natural side effect of the principal diagnosis. The CC/MCC exclusion table is organized to reflect a direct relationship between a principal diagnosis and selected secondaries.

Because the ICD-10-CM codes are non-contiguous and do not lend themselves well to defining ranges of codes, an index number is associated with each diagnosis and the CC/MCC exclusion table is constructed entirely from those index numbers.

To determine whether a secondary should be considered a CC/MCC, the executor accesses the CC/MCC table, using the principal diagnosis CC/MCC exclusion category as the key each time a secondary flagged as CC/MCC is encountered.

- If no entry is found for the exclusion category, that means that there are no exclusions and the secondary is considered a CC/MCC code.
- If an entry is found, then the secondary is excluded as a CC/MCC.

## Testing for the OTHOR condition

This test is similar in logic to the test for the ONLY conditions, except that it tests for procedures in addition to the O.R. criteria in the DRG mask. When the DRG mask indicates that other O.R. procedures must be present, the executor loops through the O.R. procedures from the record, making decisions as follows:

1. The O.R. portion of the DRG mask is moved to a work area.
2. The work mask is ANDed with the mask of the saved O.R. procedure.
  - If the result of the ANDing is zero, this indicates that the procedure is other than the specific procedure required (e.g., T&A) and therefore satisfies the other O.R. criteria. When that occurs, looping ceases and processing continues for the DRG.
  - Otherwise, the loop continues until a procedure satisfies the other condition. If all saved procedures are exhausted without finding one that satisfies the other condition, then processing for that DRG is ended.

## Testing for illogical principal diagnosis

When a DRG has been matched, and the DRG number is 999, the cause is an illogical principal diagnosis. To indicate this, the return code is changed to 6.

## Testing for multiple significant trauma

The principal diagnosis is tested to see if it is a trauma code. If it is, processing continues to test for multiple significant trauma. Otherwise, no further trauma testing is done.

To qualify as multiple significant trauma, two significant trauma codes from different body sites must be present. The diagnosis mask contains special trauma indicators, with each body site trauma represented by a different flag.

The mask of the first diagnosis (either principal or secondary) that is flagged as a significant trauma is saved. The mask of each subsequent diagnosis that is also flagged is compared with the initial saved mask. If they are not the same, the record is flagged as a multiple significant trauma episode. If they are the same, the next diagnosis is tested until the multiple condition is satisfied or the diagnoses are exhausted.

## Finding codes that affect Initial DRG assignment

After the DRG has been determined, the grouper executor analyzes the saved diagnosis and procedure masks, comparing them against the masks which were used to determine MDC and DRG. Codes which were necessary for the determination of the MDC/DRG are flagged with an "affect flag."

## Final DRG

If no Hospital Acquired Conditions (HACs) are found on the record, then the initial DRG becomes the final DRG. Otherwise, the record is re-grouped demoting the HAC secondary diagnosis which may or may not change the DRG assignment based on what DRG it was initially assigned to, and/or the presence of other codes that are CCs or MCCs.

## Executor ABEND codes

There is one ABEND (abnormal end of job) code that can be generated by the executor, standard version only.

**Table 29. ABEND codes generated by the executor-standard version**

Code	Description
108	Not able to GETMAIN a work area of sufficient size

The alternate interface does not contain any ABEND macros.



# Appendix A: Grouping results for the test database

The following is a partial listing of the output produced by the grouper utility program (DT370UT). The program's printout is a distribution of record counts by final DRG, MDC, and return code (RTC), respectively. The test database used a POA indicator of Z. There were no POAs assigned to the diagnosis codes. The printout of counts from your test run may differ in appearance from what is shown in the appendix, but the content should be the same if the test is successful. Some editing was done in order to fit the text into this manual.

The test, when performed on an IBM Z14 Model 3906-M01(608) computer, used 192K of virtual storage, and took less than 1 CPU second.

## COUNTS BY DRG

1	10	51	0	101	10	151	10	201	10	251	10	301	10	351	10
2	10	52	10	102	10	152	10	202	10	252	10	302	10	352	10
3	10	53	10	103	10	153	10	203	10	253	10	303	10	353	10
4	10	54	10	104	0	154	10	204	10	254	10	304	10	354	10
5	10	55	10	105	0	155	10	205	10	255	10	305	10	355	10
6	10	56	10	106	0	156	10	206	10	256	10	306	10	356	10
7	10	57	10	107	0	157	10	207	10	257	10	307	10	357	10
8	10	58	10	108	0	158	10	208	10	258	10	308	10	358	10
9	0	59	10	109	0	159	10	209	0	259	10	309	10	359	0
10	10	60	10	110	0	160	0	210	0	260	10	310	10	360	0
11	10	61	10	111	0	161	0	211	0	261	10	311	10	361	0
12	10	62	10	112	0	162	0	212	0	262	10	312	10	362	0
13	10	63	10	113	10	163	10	213	0	263	10	313	10	363	0
14	10	64	10	114	10	164	10	214	0	264	10	314	10	364	0
15	0	65	10	115	10	165	10	215	10	265	10	315	10	365	0
16	10	66	10	116	10	166	10	216	10	266	10	316	10	366	0
17	10	67	10	117	10	167	10	217	10	267	10	317	0	367	0
18	0	68	10	118	0	168	10	218	10	268	10	318	0	368	10
19	0	69	10	119	0	169	0	219	10	269	10	319	10	369	10
20	10	70	10	120	0	170	0	220	10	270	10	320	10	370	10
21	10	71	10	121	10	171	0	221	10	271	10	321	0	371	10
22	10	72	10	122	10	172	0	222	10	272	10	322	0	372	10
23	10	73	10	123	10	173	0	223	10	273	10	323	0	373	10
24	10	74	10	124	10	174	0	224	10	274	10	324	0	374	10
25	10	75	10	125	10	175	10	225	10	275	0	325	0	375	10
26	10	76	10	126	0	176	10	226	10	276	0	326	10	376	10
27	10	77	3	127	0	177	10	227	10	277	0	327	10	377	10
28	10	78	3	128	0	178	10	228	10	278	0	328	10	378	10
29	10	79	5	129	10	179	10	229	10	279	0	329	10	379	10
30	10	80	10	130	10	180	10	230	0	280	11	330	10	380	10
31	10	81	10	131	10	181	10	231	10	281	11	331	10	381	10
32	10	82	10	132	10	182	10	232	10	282	11	332	10	382	10

33	10		83	10		133	10		183	10		233	10		283	10		333	10		383	10	
34	10		84	10		134	10		184	10		234	10		284	10		334	10		384	10	
35	10		85	10		135	10		185	10		235	10		285	10		335	10		385	10	
36	10		86	10		136	10		186	10		236	10		286	10		336	10		386	10	
37	10		87	10		137	10		187	10		237	0		287	10		337	10		387	10	
38	10		88	10		138	10		188	10		238	0		288	10		338	10		388	10	
39	10		89	10		139	10		189	10		239	10		289	10		339	10		389	10	
40	10		90	10		140	0		190	10		240	10		290	10		340	10		390	10	
41	10		91	10		141	0		191	10		241	10		291	10		341	10		391	10	
42	10		92	10		142	0		192	10		242	10		292	10		342	10		392	10	
43	0		93	10		143	0		193	10		243	10		293	10		343	10		393	10	
44	0		94	10		144	0		194	10		244	10		294	10		344	10		394	10	
45	0		95	10		145	0		195	10		245	10		295	10		345	10		395	10	
46	0		96	10		146	10		196	10		246	10		296	10		346	10		396	0	
47	0		97	10		147	10		197	10		247	10		297	10		347	10		397	0	
48	0		98	10		148	10		198	10		248	10		298	10		348	10		398	0	
49	0		99	10		149	10		199	10		249	10		299	10		349	10		399	0	
50	0		100	10		150	10		200	10		250	10		300	10		350	10		400	0	

COUNTS BY DRG

401	0		451	0		501	10		551	10		601	10		651	0		701	0		751	0	
402	0		452	0		502	10		552	10		602	10		652	10		702	0		752	0	
403	0		453	10		503	10		553	10		603	10		653	10		703	0		753	0	
404	0		454	10		504	10		554	10		604	10		654	10		704	0		754	10	
405	10		455	10		505	10		555	10		605	10		655	10		705	0		755	10	
406	10		456	10		506	10		556	10		606	10		656	10		706	0		756	10	
407	10		457	10		507	10		557	10		607	10		657	10		707	10		757	10	
408	10		458	10		508	10		558	10		608	0		658	10		708	10		758	10	
409	10		459	10		509	10		559	10		609	0		659	10		709	10		759	10	
410	10		460	10		510	10		560	10		610	0		660	10		710	10		760	10	
411	10		461	10		511	10		561	10		611	0		661	10		711	10		761	10	
412	10		462	10		512	10		562	10		612	0		662	10		712	10		762	0	
413	10		463	10		513	10		563	10		613	0		663	10		713	10		763	0	
414	10		464	10		514	10		564	10		614	10		664	10		714	10		764	0	
415	10		465	10		515	10		565	10		615	10		665	10		715	10		765	0	
416	10		466	10		516	10		566	10		616	10		666	10		716	10		766	0	
417	10		467	10		517	10		567	0		617	10		667	10		717	10		767	0	
418	10		468	10		518	10		568	0		618	10		668	10		718	10		768	10	
419	10		469	10		519	10		569	0		619	10		669	10		719	0		769	10	
420	10		470	10		520	10		570	10		620	10		670	10		720	0		770	10	
421	10		471	10		521	0		571	10		621	10		671	10		721	0		771	0	
422	10		472	10		522	0		572	10		622	10		672	10		722	10		772	0	
423	10		473	10		523	0		573	10		623	10		673	10		723	10		773	0	
424	10		474	10		524	0		574	10		624	10		674	10		724	10		774	0	
425	10		475	10		525	0		575	10		625	10		675	10		725	10		775	0	
426	0		476	10		526	0		576	10		626	10		676	0		726	10		776	10	
427	0		477	10		527	0		577	10		627	10		677	0		727	10		777	0	
428	0		478	10		528	0		578	10		628	10		678	0		728	10		778	0	
429	0		479	10		529	0		579	10		629	10		679	0		729	10		779	10	
430	0		480	10		530	0		580	10		630	10		680	0		730	10		780	0	

431	0	481	10	531	0	581	10	631	0	681	0	731	0	781	0
432	10	482	10	532	0	582	10	632	0	682	10	732	0	782	0
433	10	483	10	533	10	583	10	633	0	683	10	733	0	783	10
434	10	484	0	534	10	584	10	634	0	684	10	734	10	784	10
435	10	485	10	535	10	585	10	635	0	685	0	735	10	785	10
436	10	486	10	536	10	586	0	636	0	686	10	736	10	786	10
437	10	487	10	537	10	587	0	637	10	687	10	737	10	787	10
438	10	488	10	538	10	588	0	638	10	688	10	738	10	788	10
439	10	489	10	539	10	589	0	639	10	689	10	739	10	789	10
440	10	490	0	540	10	590	0	640	10	690	10	740	10	790	10
441	10	491	0	541	10	591	0	641	10	691	0	741	10	791	10
442	10	492	10	542	10	592	10	642	10	692	0	742	10	792	10
443	10	493	10	543	10	593	10	643	10	693	10	743	10	793	10
444	10	494	10	544	10	594	10	644	10	694	10	744	10	794	10
445	10	495	10	545	10	595	10	645	10	695	10	745	10	795	10
446	10	496	10	546	10	596	10	646	0	696	10	746	10	796	10
447	0	497	10	547	10	597	10	647	0	697	10	747	10	797	10
448	0	498	10	548	10	598	10	648	0	698	10	748	10	798	10
449	0	499	10	549	10	599	10	649	0	699	10	749	10	799	10
450	0	500	10	550	10	600	10	650	0	700	10	750	10	800	10

COUNTS BY DRG

801	10	826	10	851	0	876	10	901	10	926	0	951	10	976	10
802	10	827	10	852	0	877	0	902	10	927	10	952	0	977	10
803	10	828	10	853	10	878	0	903	10	928	10	953	0	978	0
804	10	829	10	854	10	879	0	904	10	929	10	954	0	979	0
805	10	830	10	855	10	880	10	905	10	930	0	955	10	980	0
806	10	831	10	856	10	881	10	906	10	931	0	956	10	981	10
807	10	832	10	857	10	882	10	907	10	932	0	957	10	982	10
808	10	833	10	858	10	883	10	908	10	933	10	958	10	983	10
809	10	834	10	859	0	884	10	909	10	934	10	959	10	984	0
810	10	835	10	860	0	885	10	910	0	935	10	960	0	985	0
811	10	836	10	861	0	886	10	911	0	936	0	961	0	986	0
812	10	837	10	862	10	887	10	912	0	937	0	962	0	987	10
813	10	838	10	863	10	888	0	913	10	938	0	963	10	988	10
814	10	839	10	864	10	889	0	914	10	939	10	964	10	989	10
815	10	840	10	865	10	890	0	915	10	940	10	965	10	990	0
816	10	841	10	866	10	891	0	916	10	941	10	966	0	991	0
817	10	842	10	867	10	892	0	917	10	942	0	967	0	992	0
818	10	843	10	868	10	893	0	918	10	943	0	968	0	993	0
819	10	844	10	869	10	894	10	919	10	944	0	969	10	994	0
820	10	845	10	870	10	895	10	920	10	945	10	970	10	995	0
821	10	846	10	871	10	896	10	921	10	946	10	971	0	996	0
822	10	847	10	872	10	897	10	922	10	947	10	972	0	997	0
823	10	848	10	873	0	898	0	923	10	948	10	973	0	998	10
824	10	849	10	874	0	899	0	924	0	949	10	974	10	999	37
825	10	850	0	875	0	900	0	925	0	950	10	975	10		

COUNTS BY MDC

0 37

1	786
2	131
3	262
4	404
5	983
6	630
7	379
8	991
9	320
10	270
11	412
12	210
13	250
14	240
15	70
16	153
17	273
18	170
19	90
20	40
21	200
22	60
23	100
24	80
25	80

COUNTS BY RTC

0	7584
1	10
2	0
3	0
4	10
5	7
6	0
7	10
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0

TOTAL RECORDS PROCESSED 7621

# Index

<b>A</b>	
ABEND codes.....	57
admission date control statement.....	42
affect flag .....	57
age	
control statement.....	41
alternate interface .....	50
re 59	
ancillary buffer .....	19
ANYCOMB condition	
DRG determination.....	55
<b>B</b>	
buffer control statement .....	44
<b>C</b>	
CC exclusion subroutine	
DRG determination.....	55
COBOL	
test interface program.....	25
using the grouper with .....	47
code descriptions returned by the grouper	
.....	14
control statement	
admission date.....	42
age .....	41
buffer (BUF) .....	44
diagnosis flags (DFL) .....	44
discharge date .....	42
discharge diagnosis.....	40
discharge status.....	42
DRG .....	44
grouper flags.....	44
grouper utility .....	40
MDC .....	43
output .....	43
present on admission .....	42
procedure .....	41
procedure dates.....	43
procedure flags (SFL) .....	44
return code .....	43
sex.....	42
<b>D</b>	
data format requirements .....	9
data formats.....	10
diagnosis .....	51
diagnosis flags control statement.....	44
discharge date control statement.....	42
discharge diagnosis control statement .....	40
discharge status control statement .....	42
DRG control statement .....	44
DRG determination .....	54
ANYCOMB condition.....	55
CC exclusion subroutine .....	55
illogical principal diagnosis .....	56
multiple significant trauma.....	56
ONLY DX condition.....	55
ONLY surgery condition .....	54
OTHOR condition .....	56
OWISE condition .....	55
<b>E</b>	
Executor processing, diagnosis and	
procedure buffer .....	51
Executor, grouper .....	9
<b>F</b>	
flags returned by the grouper.....	14
folder contents, grouper .....	21
format requirements for data.....	10
<b>G</b>	
grouper	
code descriptions returned .....	43
executor .....	9
folder contents .....	21
implementation .....	9
utility program .....	9
version number.....	19
grouper executor	
ABEND codes .....	57
DRG determination.....	54

record mask .....	53	MDC control statement .....	43
grouper flags (GFL) control statement.....	44	multiple significant trauma, DRG	
grouper program.....	25	determination.....	56
grouper return code.....	14		
grouper subroutines		<b>O</b>	
input.....	48	object lib.....	24
output .....	50	ONLY DX condition, DRG determination...	55
grouper tables loader program.....	25	ONLY surgery condition, DRG determination	
grouper utility		.....	54
control statement .....	40	OTHOR condition, DRG determination .....	56
interface program.....	25	output	
link-editing .....	39	control statement.....	43
running the program .....	45	grouper .....	13
using.....	39	grouper subroutines .....	50
		OWISE condition, DRG determination .....	55
<b>H</b>			
higher level languages, COBOL .....	47	<b>P</b>	
		present on admission control statement..	42
<b>I</b>		procedure buffer executor processing .....	51
ICD-10-CM/PCS grouper		procedure control statement.....	41
CC/MCC exclusions .....	35	procedure dates control statement.....	43
diagnosis pattern table .....	31	procedure flags (SFL) control statement ..	44
diagnosis table .....	31		
grouper logic.....	30	<b>R</b>	
list of attributes .....	29	record layout	
procedure cluster definitions .....	33	DRGDSC3.....	28
procedure clusters.....	30	DRGDSC4.....	29
procedure pattern table .....	33	MDCDSC.....	27
procedure table .....	32	test database .....	26
ICD-10-CM/PCS grouper attributes		record mask, construction .....	53
diagnosis .....	31	return code	
procedure .....	32	control statement.....	43
ICD-10-CM/PCS grouper tables.....	29	test database .....	59
illogical principal diagnosis.....	56		
information returned by the grouper .....	11	<b>S</b>	
initial DRG assignment .....	57	sample JCL	
interface, alternate .....	50	creating the grouper utility load module	
		.....	39
<b>L</b>		grouping test database .....	45
link-editing, grouper utility .....	39	sex .....	42
		source library, grouper program.....	25
<b>M</b>		SYSIN stream .....	39
main control program .....	25		

## T

test database	
record counts.....	59
record layout .....	26
return code .....	59