

*****PSEUDOCODE FOR MDS-RCA STATE OF MAINE GROUP*****
 *****Version 2000*****

*****Last update September 2002

**** This is the pseudocode to be used by vendors – it does not include the code for older version

** This document contains a long documentation section followed by the actual pseudocode.

 **CALCULATION TYPES USED IN THIS PSEUDOCODE:

** This code allows for 15 group classifications, using the Hierarchical type and the Index maximizing calculation types.

** For Hierarchical type, a resident is placed in the first group for which they qualify. The order of the groups in this grouper is Impaired, Complex, Behavioral Health, and Physical.

** For the Index maximizing calculation type, there is no order of precedence for the groups. For this latter method, all groups for which a resident would qualify are determined, and the resident is placed in the group with the highest CMI (Case Mix Index). For the Index maximizing method a CMI SET is provided

- ** RugGroup *2 alpha characters and the numeric character 1
- ** Hier order - 5 hierarchical group (Impaired, Complex, Behavioral Health, Physical and Not Classified)
- ** Order - rug group order by Hierarchical and further subdivided by ADL scores
- ** MedicaidWeight - CMI (Case Mix Index)
- ** Short description - description of group
- ** Hier - the 5 hierarchical groups

** the calculation type (hierarchical or index maximizing) used is indicated by local variables:

- ** strClinicalGroup indicates the hierarchical rug group, dblClinicMedicaidWgt is the hierarchical CMI
- ** strPaymentGroup indicates index maximizing rug group, dblPayMedicaidWgt is the index maximizing CMI

** Rug Group	Hier	HierOrder	Order	Short Description	MedicaidWgt
** IC1	IMPAIRED	1	1	IMPAIRED 15-28	2.25
** IB1	IMPAIRED	1	2	IMPAIRED 12-14	1.568
** IA1	IMPAIRED	1	3	IMPAIRED 0-11	1.144
** CD1	COMPLEX	2	4	COMPLEX 12+	1.944
** CC1	COMPLEX	2	5	COMPLEX 7-11	1.593
** CB1	COMPLEX	2	6	COMPLEX 2-6	1.205
** CA1	COMPLEX	2	7	COMPLEX 0-1	0.938
** MC1	BEHAVIORAL HEALTH	3	8	BEHAVIORAL HEALTH 16+	1.916
** MB1	BEHAVIORAL HEALTH	3	9	BEHAVIORAL HEALTH 5-15	1.377
** MA1	BEHAVIORAL HEALTH	3	10	BEHAVIORAL HEALTH 0-4	0.98
** PD1	PHYSICAL	4	11	PHYSICAL 11+	1.418
** PC1	PHYSICAL	4	12	PHYSICAL 8-10	1.019
** PB1	PHYSICAL	4	13	PHYSICAL 4-7	1.004
** PA1	PHYSICAL	4	14	PHYSICAL 0-3	0.731
** BC1	NOT CLASSIFIED	5	15	NOT CLASSIFIED	0.731

 ** SYNTAX USED IN THIS PSEUDOCODE:

- ** 1. All lines with asteriks (**) as the first nonblank characters in the line are documentation or comment lines. All command lines start with characters other than asteriks
- ** 2. Any command line which ends with _ is continued on the next line. An underscore _ is the continuation character and this is the only punctuation used in command lines.

- ** 3. All variables are represented by small letters and Capital letters to distinguish between words. While command words are represented by all caps (e.g. - IF, ENDIF)
- ** 4. MDS-RCA fields will be represented with a prefix of "r_". (e.g. r_B3 for the B3 item from the MDS-RCA).
*8 ALL MDS-RCA VARIABLES ARE REQUIRED TO BE CHARACTER VARIABLES IN THIS PSEUDOCODE.
** The MDS-RCA variable correspond to the labels on the MDS-RCA form.
- ** 5. All local variables are presented with a prefix of "str" for character variable, "int" for an integer numeric variable, "dbl" for a double numeric variable.
- ** 6. The only command words and structure used in this pseudocode are:
 - ** a. RETURN (this command is the last command and indicates the end of the code)
 - ** b. IF <logical condition> THEN
** <statement> executed IF condition is TRUE
** ENDIF
 - ** c. IF <logical condition> THEN
** <statement> executed IF condition1 is TRUE
** ELSEIF <logical condition>
** <statement> executed IF condition2 is TRUE
** ENDIF
 - ** d. Logical operators used in logical conditions are
 - ** = equal
 - ** > greater than
 - ** < less than
 - ** >= greater than or equal to
 - ** <= less than or equal to
 - ** NOT not equal to -- less than or greater than
 - ** e. Relational operators used in logical conditions are
 - ** AND logical and
 - ** OR logical or
 - ** f. VAL(str_var) The VAL function returns the numeric value for a character variable (str_var)
 - ** g. NEXT used to loop through records

** PROCESSING REQUIRED BEFORE EXECUTION OF PSEUDOCODE:

- ** 1. Run edits on MDS-RCA records. All fields with invalid data must have been replaced with "**".
 - ** 2. All mds-rca records will be stored with all values from all mds-rca fields used in this pseudocode.
 - ** 3. Records with invalid values (*) in ANY required MDS-RCA variable will be assigned (BC1) and the calculation code in this document is skipped.
-

** DECLARE RUGiii VARIABLES

```
strClinicalGroup As String
strPaymentGroup As String
dblPayMedicaidWgt As Double
dblClinicMedicaidWgt As Double
bolComplex As Boolean
bolUnclassified as boolean
bolMood As Boolean
intADL_Bed As Integer
intADL_Transfer As Integer
intADL_Locomotion As Integer
intADL_Dressing As Integer
intADL_Eating As Integer
intADL_Toilet As Integer
intADL_Hygiene As Integer
intADL As Integer
intE1Count As Integer
```

intP2Count As Integer
strCriteria As String
intRsCount As Integer ** record count
intI As Integer **Loop counter
intI = 1

*** MDS-RCA FIELDS USED IN GROUPER *****

** Cognitive Skill for daily decision making
** B3 Made decisions regarding tasks of daily IIFe

** Indicators of Anxiety, depression and sad mood

** E1a Negative statements
** E1b Repetitive questions
** E1c Repetitive verbalizations
** E1d Persistent anger with self or others
** E1e Self deprecation
** E1f Unrealistic fears
** E1g Recurrent statements that something bad will happen
** E1h Repetitive health complaints
** E1i Repetitive anxious complaints/concerns
** E1j Unpleasant mood in morning
** E1k Insomnia/change in sleep pattern
** E1l Sad, pained, worried facial expressions
** E1m Crying, tearfulness
** E1n Repetitive physical movements
** E1o Withdrawal from activities of interest
** E1p Reduced social intervention
** E1q Inflated self-worth, arrogance
** E1r Excited behavior, motor excitation

** Physical functioning - ADL Self-performance

** G1aa Bed mobility
** G1ba Transfer
** G1ca Locomotion out of room
** G1da Dressing
** G1ea Eating
** G1fa Toilet use
** G1ga Personal hygiene

** Diagnoses

** I1a Diabetes Mellitus
** I1r Aphasia
** I1s Cerebral Palsy
** I1v Hemiplegia/hemiparesis
** I1w Mutiple sclerosis
** I1z Quadriplegia
** I1ww Explicit terminal prognosis

** Health conditions

** J1e Delusions
** J1f Hallucinations

- ** Skin condition
 - ** M1b Burns
- ** Skin condition - Ulcers
 - ** M2a Ulcers - Stage 1
 - ** M2b Ulcers - Stage 2
 - ** M2c Ulcers - Stage 3
 - ** M2d Ulcers - Stage 4
- ** Injections
 - ** O3 Injections = 30
- ** Special treatments and procedures
- ** Special care
 - ** P1aa Chemotherapy or radiation
 - ** P1ab Oxygen
- ** Therapy
 - ** P1bda Respiratory therapy
- ** Intervention Programs for mood, behavior, cognitive loss
 - ** P2a Special behavior symptom evaluation program
 - ** P2b Special behavior management program
 - ** P2c Evaluation by licensed mental health specialist in last 90 days
 - ** P2d Group therapy
 - ** P2e Resident-specific deliberate changes in environment
 - ** P2f Reorientation - cueing
 - ** P2g Validation/redirection
 - ** P2h Crisis intervention in facility
 - ** P2i Crisis stabilization unit in last 90 days
 - ** P2j Other
- ** Need for ongoing monitoring
 - ** P3a Acute physical or psychiatric condition
 - ** P3b New treatment/medication
- ** Physician Orders
 - ** P10 Physician orders in last 14 days

** -----
 *****END OF DOCUMENTATION*****

*****BEGINNING OF PSEUDOCODE*****

intRsCount = number of records to be grouped
 move to the first record

** loop through records one at a time and execute code

FOR <each record>

** assigns starting values to rugiii variables

```
strFieldPutInComplex = ""
strFieldPutInUnclassified = ""
strFieldPutInBehavioral = ""
strFieldPutInImpaired = ""
strClinicalGroup = "BC1"
strPaymentGroup = "BC1"
dblPayMedicaidWgt = -999.9
dblClinicMedicaidWgt = -999.9
bolComplex = FALSE
bolMood = FALSE
intADL_Bed = 0
intADL_Transfer = 0
intADL_Locomotion = 0
intADL_Dressing = 0
intADL_Eating = 0
intADL_Toilet = 0
intADL_Hygiene = 0
intADL = 0
intE1Count = 0
intP2Count = 0
```

** Catch stars for unclassified here (records with invalid values)

```
IF <r_B3> = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1a = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1b = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1c = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1d = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1e = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1f = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1g = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1h = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1i = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1j = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1k = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1l = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1m = "*" THEN
  bolUnclassified = TRUE
ELSEIF r_E1n = "*" THEN
```

bolUnclassified = TRUE
ELSEIF r_E1o = "*" THEN
bolUnclassified = TRUE
ELSEIF r_E1p = "*" THEN
bolUnclassified = TRUE
ELSEIF r_E1q = "*" THEN
bolUnclassified = TRUE
ELSEIF r_E1r = "*" THEN
bolUnclassified = TRUE
ELSEIF r_G1aa = "*" THEN
bolUnclassified = TRUE
ELSEIF r_G1ba = "*" THEN
bolUnclassified = TRUE
ELSEIF r_G1ca = "*" THEN
bolUnclassified = TRUE
ELSEIF r_G1da = "*" THEN
bolUnclassified = TRUE
ELSEIF r_G1ea = "*" THEN
bolUnclassified = TRUE
ELSEIF r_G1fa = "*" THEN
bolUnclassified = TRUE
ELSEIF r_G1ga = "*" THEN
bolUnclassified = TRUE
ELSEIF r_I1a = "*" THEN
bolUnclassified = TRUE
ELSEIF r_I1r = "*" THEN
bolUnclassified = TRUE
ELSEIF r_I1s = "*" THEN
bolUnclassified = TRUE
ELSEIF r_I1v = "*" THEN
bolUnclassified = TRUE
ELSEIF r_I1w = "*" THEN
bolUnclassified = TRUE
ELSEIF r_I1z = "*" THEN
bolUnclassified = TRUE
ELSEIF r_I1ww = "*" THEN
bolUnclassified = TRUE
ELSEIF r_J1e = "*" THEN
bolUnclassified = TRUE
ELSEIF r_J1f = "*" THEN
bolUnclassified = TRUE
ELSEIF r_M1b = "*" THEN
bolUnclassified = TRUE
ELSEIF r_M2a = "*" THEN
bolUnclassified = TRUE
ELSEIF r_M2b = "*" THEN
bolUnclassified = TRUE
ELSEIF r_M2c = "*" THEN
bolUnclassified = TRUE
ELSEIF r_M2d = "*" THEN
bolUnclassified = TRUE
ELSEIF r_O4ag = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P1aa = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P1ab = "*" THEN

```
bolUnclassified = TRUE
ELSEIF r_P1bda = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P2a = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P2b = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P2c = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P2d = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P2e = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P2f = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P2g = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P2h = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P2i = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P2j = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P3a = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P3b = "*" THEN
bolUnclassified = TRUE
ELSEIF r_P10 = "***" Or r_P10 = "*" THEN
bolUnclassified = TRUE
End IF
```

```
IF bolUnclassified = TRUE THEN
strClinical Group = "BC1"
strPaymentGroup = "BC1"
dblClinicMedicaidWgt = CMI for BC1 rug group
dblPayMedicaidWgt = CMI for BC1 rug group
ELSE ' if all valid values then do the following
```

```
** Calculate ADL scores
```

```
IF r_G1aa = "0" THEN
intADL_Bed = 0
ELSEIF r_G1aa = "1" THEN
intADL_Bed = 1
ELSEIF r_G1aa = "2" THEN
intADL_Bed = 2
ELSEIF r_G1aa = "3" THEN
intADL_Bed = 3
ELSEIF (r_G1aa = "4" Or r_G1aa = "8") THEN
intADL_Bed = 4
End IF
```

**ADL SCORE FOR TRANSFER

```
IF r_G1ba = "0" THEN
  intADL_Transfer = 0
ELSEIF r_G1ba = "1" THEN
  intADL_Transfer = 1
ELSEIF r_G1ba = "2" THEN
  intADL_Transfer = 2
ELSEIF r_G1ba = "3" THEN
  intADL_Transfer = 3
ELSEIF (r_G1ba = "4" Or r_G1ba = "8") THEN
  intADL_Transfer = 4
End IF
```

**ADL SCORE FOR LOCOMOTION

```
IF r_G1ca = "0" THEN
  intADL_Locomotion = 0
ELSEIF r_G1ca = "1" THEN
  intADL_Locomotion = 1
ELSEIF r_G1ca = "2" THEN
  intADL_Locomotion = 2
ELSEIF r_G1ca = "3" THEN
  intADL_Locomotion = 3
ELSEIF (r_G1ca = "4" Or r_G1ca = "8") THEN
  intADL_Locomotion = 4
End IF
```

**ADL SCORE FOR DRESSING

```
IF r_G1da = "0" THEN
  intADL_Dressing = 0
ELSEIF r_G1da = "1" THEN
  intADL_Dressing = 1
ELSEIF r_G1da = "2" THEN
  intADL_Dressing = 2
ELSEIF r_G1da = "3" THEN
  intADL_Dressing = 3
ELSEIF (r_G1da = "4" Or r_G1da = "8") THEN
  intADL_Dressing = 4
End IF
```

**ADL SCORE FOR EATING

```
IF r_G1ea = "0" THEN
  intADL_Eating = 0
ELSEIF r_G1ea = "1" THEN
  intADL_Eating = 1
ELSEIF r_G1ea = "2" THEN
  intADL_Eating = 2
ELSEIF r_G1ea = "3" THEN
  intADL_Eating = 3
ELSEIF (r_G1ea = "4" Or r_G1ea = "8") THEN
  intADL_Eating = 4
End IF
```

**ADL SCORE FOR TOILET

```
IF r_G1fa = "0" THEN
  intADL_Toilet = 0
ELSEIF r_G1fa = "1" THEN
  intADL_Toilet = 1
ELSEIF r_G1fa = "2" THEN
  intADL_Toilet = 2
ELSEIF r_G1fa = "3" THEN
  intADL_Toilet = 3
ELSEIF (r_G1fa = "4" Or r_G1fa = "8") THEN
  intADL_Toilet = 4
End IF
```

**ADL SCORE FOR HYGIENE

```
IF r_G1ga = "0" THEN
  intADL_Hygiene = 0
ELSEIF r_G1ga = "1" THEN
  intADL_Hygiene = 1
ELSEIF r_G1ga = "2" THEN
  intADL_Hygiene = 2
ELSEIF r_G1ga = "3" THEN
  intADL_Hygiene = 3
ELSEIF (r_G1ga = "4" Or r_G1ga = "8") THEN
  intADL_Hygiene = 4
End IF
```

** get total ADL SCORE

```
intADL = intADL_Bed + intADL_Transfer + intADL_Locomotion + intADL_Dressing
intADL = intADL + intADL_Eating + intADL_Toilet + intADL_Hygiene
```

**SET COMPLEX FLAG

```
IF r_I1a = "1" And r_O4ag = "7" THEN
  bolComplex = TRUE
ELSEIF r_I1r = "1" Or r_I1s = "1" Or r_I1v = "1" THEN
  bolComplex = TRUE
ELSEIF r_I1w = "1" Or r_I1ww = "1" Or r_I1z = "1" _
  Or r_M1b = "1" THEN
  bolComplex = TRUE
ELSEIF r_P3a = "1" Or r_P3a = "2" _
  Or r_P3a = "3" THEN
  bolComplex = TRUE
ELSEIF r_P3b = "1" Or r_P3b = "2" _
  Or r_P3b = "3" THEN
  bolComplex = TRUE
ELSEIF r_P1aa = "1" Or r_P1ab = "1" THEN
```

```
bolComplex = TRUE
ELSEIF VAL(r_P1bda) >= 5 Or VAL(r_P10) >= 4 THEN
bolComplex = TRUE
ELSEIF r_M2a > "0" Or r_M2b > "0" Or r_M2c > "0" Or r_M2d > "0" THEN
bolComplex = TRUE
End IF
```

```
** end SET complex flag
```

```
**Calculate BEHAVIOR COUNTS
```

```
IF r_E1a > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1b > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1c > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1d > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1e > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1f > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1g > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1h > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1i > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1j > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1k > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1l > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1m > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1n > "0" THEN
intE1Count = intE1Count + 1
End IF
IF r_E1o > "0" THEN
intE1Count = intE1Count + 1
End IF
```

```
IF r_E1p > "0" THEN
    intE1Count = intE1Count + 1
End IF
IF r_E1q > "0" THEN
    intE1Count = intE1Count + 1
End IF
IF r_E1r > "0" THEN
    intE1Count = intE1Count + 1
End IF
```

```
IF r_P2a = "1" THEN
    intP2Count = intP2Count + 1
End IF
IF r_P2b = "1" THEN
    intP2Count = intP2Count + 1
End IF
IF r_P2c = "1" THEN
    intP2Count = intP2Count + 1
End IF
IF r_P2d = "1" THEN
    intP2Count = intP2Count + 1
End IF
IF r_P2e = "1" THEN
    intP2Count = intP2Count + 1
End IF
IF r_P2f = "1" THEN
    intP2Count = intP2Count + 1
End IF
IF r_P2g = "1" THEN
    intP2Count = intP2Count + 1
End IF
IF r_P2h = "1" THEN
    intP2Count = intP2Count + 1
End IF
```

```
IF r_P2i = "1" THEN
    intP2Count = intP2Count + 1
End IF
```

```
IF r_P2j = "1" THEN
    intP2Count = intP2Count + 1
End IF
```

** sets mood indicated to TRUE if condition is met

```
IF intE1Count > 1 Or intP2Count > 2 THEN
    bolMood = TRUE
End IF
```

```
IF r_J1e = "1" THEN
    bolMood = TRUE
End IF
```

```
IF r_J1f = "1" THEN
    bolMood = TRUE
End IF
```

** CALCULATE CLINICAL GROUP and PAYMENT GROUP

** checks if B3 equals 3 then record falls into the Impaired Hier group for the Hierarchical Group

IF r_B3 = "3" THEN

 ** checks ADL score to determine which of the Impair subgroups will the record fall into

 IF intADL >= 15 THEN

 strClinicalGroup = "IC1"

 strPaymentGroup = "IC1"

 dblPayMedicaidWgt = CMI for the IC1 group

 dblClinicMedicaidWgt = CMI for the IC1 group

 ELSEIF intADL >= 12 And intADL <= 14 THEN

 strClinicalGroup = "IB1"

 strPaymentGroup = "IB1"

 dblPayMedicaidWgt = CMI for the IB1 group

 dblClinicMedicaidWgt = CMI for the IB1 group

 ELSEIF intADL >= 0 And intADL <= 11 THEN

 strClinicalGroup = "IA1"

 strPaymentGroup = "IA1"

 dblPayMedicaidWgt = CMI for the IA1 group

 dblClinicMedicaidWgt = CMI for the IA1 group

 End IF ** intADL >=15

End IF ** b3 = 3

** checks for complex flag if TRUE then using ADL score determines which of the Complex subgroup the record falls into
** for the Hierarchical Group then checks the CMI to see if its lower than the current CMI if it is then the dblPayMedicaidWgt
** is set to the current CMI

IF bolComplex = TRUE THEN

 IF intADL >= 12 THEN

 IF strClinicalGroup = "BC1" THEN ** clinical group has not been previously set by qualifying for a rug group thus far

 strClinicalGroup = "CD1"

 dblClinicMedicaidWgt = CMI for the CD1 rug group

 End IF

 IF dblPayMedicaidWgt < dblClinicMedicaidWgt THEN

 dblPayMedicaidWgt = CMI for the CD1 rug group

 strPaymentGroup = "CD1"

 End IF

 ELSEIF intADL >= 7 And intADL <= 11 THEN

 IF strClinicalGroup = "BC1" THEN

 strClinicalGroup = "CC1"

 dblClinicMedicaidWgt = CMI for the CC1 rug group

 End IF

 IF dblPayMedicaidWgt < CMI for the CC1 rug group THEN

 dblPayMedicaidWgt = CMI for the CC1 rug group

 strPaymentGroup = "CC1"

 End IF

```

ELSEIF intADL >= 2 And intADL <= 6 THEN
  IF strClinicalGroup = "BC1" THEN
    strClinicalGroup = "CB1"
    dblClinicMedicaidWgt = CMI for the CB1 rug group
  End IF
  IF dblPayMedicaidWgt < CMI for the CB1 rug group THEN
    dblPayMedicaidWgt = CMI for the CB1 rug group
    strPaymentGroup = "CB1"
  End IF
ELSEIF intADL >= 0 And intADL <= 1 THEN
  IF strClinicalGroup = "BC1" THEN
    strClinicalGroup = "CA1"
    dblClinicMedicaidWgt = CMI for the CA1 rug group
  End IF
  IF dblPayMedicaidWgt < CMI for the CA1 rug group THEN
    dblPayMedicaidWgt = CMI for the CA1 rug group
    strPaymentGroup = "CA1"
  End IF
End IF ** intADL >= 12
End IF ** end complex group

```

** checks for mood flag if TRUE then using ADL score determines which of the mod subgroup the record falls into for the
** Hierarchical group

```

IF bolMood = TRUE THEN
  IF intADL >= 16 THEN
    IF strClinicalGroup = "BC1" THEN
      strClinicalGroup = "MC1"
      dblClinicMedicaidWgt = CMI for the MC1 rug group
    End IF
    IF dblPayMedicaidWgt < CMI for the MC1 rug group THEN
      dblPayMedicaidWgt = CMI for the MC1 rug group
      strPaymentGroup = "MC1"
    End IF
  ELSEIF intADL >= 5 And intADL <= 15 THEN
    IF strClinicalGroup = "BC1" THEN
      strClinicalGroup = "MB1"
      dblClinicMedicaidWgt = CMI for the MB1 rug group
    End IF
    IF dblPayMedicaidWgt < CMI for the MB1 rug group THEN
      dblPayMedicaidWgt = CMI for the MB1 rug group
      strPaymentGroup = "MB1"
    End IF
  ELSEIF intADL >= 0 And intADL <= 4 THEN
    IF strClinicalGroup = "BC1" THEN
      strClinicalGroup = "MA1"
      dblClinicMedicaidWgt = CMI for the MA1 rug group
    End IF
    IF dblPayMedicaidWgt < CMI for the MA1 rug group THEN
      dblPayMedicaidWgt = CMI for the MA1 rug group
      strPaymentGroup = "MA1"
    End IF
  End IF ** intADL >= 16
End IF **end bolMood = TRUE

```

** sets to which physical group dependent on adl score

```

IF intADL >= 11 THEN
  IF strClinicalGroup = "BC1" THEN
    strClinicalGroup = "PD1"
    dblClinicMedicaidWgt = CMI for the PD1 rug group
  End IF
  IF dblPayMedicaidWgt < CMI for the PD1 rug group THEN
    dblPayMedicaidWgt = CMI for the PD1 rug group
    strPaymentGroup = "PD1"
  End IF
ELSEIF intADL >= 8 And intADL <= 10 THEN
  IF strClinicalGroup = "BC1" THEN
    strClinicalGroup = "PC1"
    dblClinicMedicaidWgt = CMI for the PC1 rug group
  End IF
  IF dblPayMedicaidWgt < CMI for the PC1 rug group THEN
    dblPayMedicaidWgt = CMI for the PC1 rug group
    strPaymentGroup = "PC1"
  End IF
ELSEIF intADL >= 4 And intADL <= 7 THEN
  IF strClinicalGroup = "BC1" THEN
    strClinicalGroup = "PB1"
    dblClinicMedicaidWgt = CMI for the PB1 rug group
  End IF
  IF dblPayMedicaidWgt < CMI for the PB1 rug group THEN
    dblPayMedicaidWgt = CMI for the PB1 rug group
    strPaymentGroup = "PB1"
  End IF
ELSEIF intADL >= 0 And intADL <= 3 THEN
  IF strClinicalGroup = "BC1" THEN
    strClinicalGroup = "PA1"
    dblClinicMedicaidWgt = CMI for the PA1 rug group
  End IF
  IF dblPayMedicaidWgt < CMI for the PA1 rug group THEN
    dblPayMedicaidWgt = CMI for the PA1 rug group
    strPaymentGroup = "PA1"
  End IF
End IF **int adl >=11
End IF ** end IF strFieldPutInUnclassified <> ""

** Update the following:
Hierarchical Type Rug Group = strClinicalGroup
Hierarchical Type CMI = dblClinicMedicaidWgt
Index type Rug Group = strPaymentGroup
Index type CMI = dblPayMedicaidWgt

** move to next record
NEXT intI

** end after last record processed
RETURN

```