



Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Département fédéral de l'intérieur

Office fédéral de la statistique
Division Santé et affaires sociales

Silvio Hasler, 23 juin 2017

Conception de plausibilisation de la statistique médicale des hôpitaux

Version 6.0.0.2

Ce rapport n'est disponible que sous forme électronique et uniquement dans les langues française et allemande: <http://www.bfs.admin.ch>

Auteur :

*Freudiger EDV-Beratung
Zeughausgasse 16
Boîte postale
3001 Berne*

Table des matières

A. Codage	4
A0. Diagnostics	4
A1. Traitements	8
B. Données administratives	11
B0. Données du patient	11
B1. Séjour hospitalier	12
B2. Soins intensifs	17
B3. Données sur les nouveau-nés	18
B4. Données psychiatriques	23
B5. MK	29
B6. Identification des réhospitalisations	34
C. Erreurs formelles	40
C0. Problèmes structurels	40
C1. Problèmes MX	42
C2. MB	44
C7. MD	46
D. Indicateurs	50
D0. Messages	50
D1. Patients	51
D2. Admission	51
D3. Séjour	52
D4. Sortie	52
D5. Diagnostics	53
D6. Traitements	53
Annexe A. Format des fichiers de métadonnées CIM/CHOP	54
Annexe B. Description des fonctions utilisées dans les tests	55

A. Codage

A0. Diagnostics

A00. Diagnostic principal

A0000 [E] Diagnostic principal : La donnée manque (cas statistique A, C)

`DIA[0].KeyLen == 0 && !isIn(MB->V0202,2,_AC) && !(MB->hasMN && strcmp(MN->V2201,'0') == 0)`

A0010 [E] Le code de diagnostic principal n'est pas autorisé.

`DIA[0].inIndex && strcmp(DIA[0].Data[ICD_VALIDMD],'2') == 0 && *MB->MD->V4801 == '1'`

A0011 [E] Le code astérisque n'est pas autorisé comme diagnostic principal

`DIA[0].inIndex && strcmp(DIA[0].Data[ICD_TYP],'2') == 0`

A0012 [E] Codes avec ! : toujours en diagnostic supplémentaire, sauf les causes externes (V01 !-Y98 !)

`DIA[0].inIndex && isExkIDia(DIA[0].Key)`

A0013 [E] Diagnostic principal : Ici, les causes externes (CIM. Chap. XX) ne sont pas autorisées

`!DIA[0].isNull && (DIA[0].Key[0] == 'V' || DIA[0].Key[0] == 'W' || DIA[0].Key[0] == 'X' || DIA[0].Key[0] == 'Y')`

A0020 [E] Un cas dont le diagnostic principal est Z758 ne doit pas être facturé.

`DIA[0].inIndex && strcmp(DIA[0].Key,'Z758') == 0 && MB->hasMD && strcmp(MB->MD->V4801,'1') == 0`

A0021 [E] Un cas dont le diagnostic principal est Z758 doit avoir le type d'admission 4 Transfert interne.

`DIA[0].inIndex && strcmp(DIA[0].Key,'Z758') == 0 && strcmp(MB->V1203,'4') != 0`

A01. Complément au diagnostic principal

A0110 [E] Le code dague n'est pas autorisé comme complément au diagnostic principal

`DIA[1].inIndex && (strcmp(DIA[1].Data[ICD_TYP],'1') == 0 || strcmp(DIA[1].Data[ICD_TYP],'4') == 0)`

A0120 [E] Complément au diagnostic principal : uniquement pour codes astérisque (*) et codes de cause externe (V01 !-Y98 !)

`DIA[1].inIndex && (strcmp(DIA[1].Data[ICD_TYP],'1') == 0 || strcmp(DIA[1].Data[ICD_TYP],'4') == 0)`

A0130 [W] Les codes de diagnostic principal T90 - T98 ne nécessitent plus de diagnostic supplémentaire V - Y.

`DIA[0].inIndex && strcmp(DIA[0].Key,'T9',2) == 0 && DIA[1].inIndex && (strcmp(DIA[1].Key,'V',1) == 0 || strcmp(DIA[1].Key,'W',1) == 0 || strcmp(DIA[1].Key,'X',1) == 0 || strcmp(DIA[1].Key,'Y',1) == 0)`

A02. Validité des diagnostics

A0200 [E] Code diagnostique (CIM-10) non valable

```
!DIA[i].isNull && !DIA[i].inIndex
```

A0201 [E] Le code CIM n'était plus valable au moment de la sortie

```
DIA[i].inIndex && strlen(DIA[i].Data[ICD_VALID_TO]) > 0 && MB->dateV1501.hasYear && MB->dateV1501.Year > atol(DIA[i].Data[ICD_VALID_TO])
```

A0202 [E] Le code CIM n'était pas encore valable au moment de la sortie

```
DIA[i].inIndex && strlen(DIA[i].Data[ICD_VALID_FROM]) > 0 && MB->dateV1501.hasYear && MB->dateV1501.Year < atol(DIA[i].Data[ICD_VALID_FROM])
```

A0210 [E] Code diagnostique (CIM-10) non terminal

```
DIA[i].inIndex && StatYear > 2009 && StatYear >= atol(DIA[i].Data[ICD_VALID_FROM]) && StatYear <= atol(DIA[i].Data[ICD_VALID_TO]) && (StatYear < atoi(DIA[i].Data[ICD_TERM_FROM]) || StatYear > atoi(DIA[i].Data[ICD_TERM_TO]))
```

A03. Regles de codage

A0310 [W] Diagnostic / Mode de sortie : Diagnostic principal léthal, le mode de sortie ne peut qu'être 'Décès'

```
!DIA[i].isNull && strcmp(MB->V1502,'5') != 0 && (strcmp(DIA[i].Key,'R95',3) == 0 || strcmp(DIA[i].Key,'R96',3) == 0 || strcmp(DIA[i].Key,'R98',3) == 0 || strcmp(DIA[i].Key,'R99',3) == 0 || strcmp(DIA[i].Key,'P95',3) == 0 || strcmp(DIA[i].Key,'O95',3) == 0 || strcmp(DIA[i].Key,'O96',3) == 0 || strcmp(DIA[i].Key,'O97',3) == 0 || strcmp(DIA[i].Key,'I461',4) == 0 || strcmp(DIA[i].Key,'P964',4) == 0 || strcmp(DIA[i].Key,'S18',3) == 0 )
```

A0320 [E] Il faut coder les bébés nés vivants avec le code diagnostique Z38

```
strcmp(MB->V0202,'A') == 0 && MB->hasMN && strcmp(MB->V1203,'3') == 0 && strcmp(MB->MN->V2201,'1') == 0 && !hasDiaLike('Z38')
```

A0321 [E] Etat à la naissance (enfants mort-nés) / diagnostics : Le code Z38.- ne peut pas être utilisé en cas de mortinaissance

```
DIA[i].inIndex && strcmp(DIA[i].Key,'Z38',3) == 0 && MB->hasMN && strcmp(MB->MN->V2201,'0') == 0
```

A0330 [E] Les codes R75, Z21, B23.0 et le groupe B20-B24 s'excluent mutuellement et ne sont pas à coder ensemble durant un même séjour hospitalier

```
DIA[i].inIndex && hasB2024 && (strcmp(DIA[i].Key,'R75',3) == 0 || strcmp(DIA[i].Key,'Z21',3) == 0 || strcmp(DIA[i].Key,'B230',4) == 0 )
```

A0341 [E] Le code Z37 doit être codé dans l'enregistrement de la mère main non dans celui de l'enfant

```
strlen(MB->V1103) > 0 && DIA[i].inIndex && strcmp(DIA[i].Key,'Z37',3) == 0 && atoi(MB->V1103) == 0
```

A0342 [E] Le code Z38 doit être codé dans l'enregistrement de l'enfant mais non dans celui de la mère

```
strlen(MB->V1103) > 0 && DIA[i].inIndex && strcmp(DIA[i].Key,'Z38',3) == 0 && atoi(MB->V1103) > 0
```

A0343 [W] Lorsque l'un des codes suivants est entré en tant que diagnostic principal ou supplémentaire, le code O09.- ! doit également figurer dans les diagnostics supplémentaires : O03-O07 ; O10-O16 ; O20-O29 ; O30-O48 ; O60-O75 ; O80-O82 ; Z34-Z37

DIA[i].inIndex && isSSDia(DIA[i].Key) && !hasO09

A0344 [E] Avec un code O80-O82, le code Z37 doit obligatoirement figurer dans les diagnostics supplémentaires

DIA[i].inIndex && (strncmp(DIA[i].Key,'O80',3) == 0 || strncmp(DIA[i].Key,'O81',3) == 0 || strncmp(DIA[i].Key,'O82',3) == 0) && !hasZ37

A0345 [E] Les codes O80, O81, O82 sont des codes pour diagnostic principal

i > 0 && DIA[i].inIndex && (strncmp(DIA[i].Key,'O80',3) == 0 || strncmp(DIA[i].Key,'O81',3) == 0 || strncmp(DIA[i].Key,'O82',3) == 0)

A0350 [W] Les diagnostics Z51.0, Z51.1, Z51.82 ne seront pas codés.

DIA[i].inIndex && (strcmp(DIA[i].Key,'Z510') == 0 || strcmp(DIA[i].Key,'Z511') == 0 || strcmp(DIA[i].Key,'Z5182') == 0)

A0361 [X] Pour une Naissance unique, un des codes suivants doit être codé : Z38.0/1/2

*V2201 == '1' && *V2202 == '1' && !(MB->MD->hasDiaLike('Z380') || MB->MD->hasDiaLike('Z381') || MB->MD->hasDiaLike('Z382'))

A0362 [X] Pour des jumeaux un des codes suivants doit être codé : Z38.3/4/5

*V2201 == '1' && *V2202 == '2' && !(MB->MD->hasDiaLike('Z383') || MB->MD->hasDiaLike('Z384') || MB->MD->hasDiaLike('Z385'))

A0363 [X] Pour une naissance multiple un des codes suivants doit être codé : Z38.6/7/8

*V2201 == '1' && atoi(V2202) > 2 && !(MB->MD->hasDiaLike('Z386') || MB->MD->hasDiaLike('Z387') || MB->MD->hasDiaLike('Z388'))

A0364 [X] Pour une Naissance unique, un des codes suivants doit être codé : Z38.0/1/2

(hasDiaLike('Z380') || hasDiaLike('Z381') || hasDiaLike('Z382')) && (!MB->hasMN || *(MN->V2201) != '1' || *(MN->V2202) != '1')

A0365 [X] Si un jumeaux est codé avec Z38.3/4/5, il doit aussi être déclare dans les suppl. nouveau-nées

(hasDiaLike('Z383') || hasDiaLike('Z384') || hasDiaLike('Z385')) && (!MB->hasMN || *(MN->V2201) != '1' || *(MN->V2202) != '2')

A0366 [X] Si une naissance multiple est codé avec Z38.6/7/8, elle doit aussi être déclaré dans les nouveau-nées (MN).

(hasDiaLike('Z386') || hasDiaLike('Z387') || hasDiaLike('Z388')) && (!MB->hasMN || *(MN->V2201) != '1' || atoi(MN->V2202) < 3)

A04. Plausibilité des diagnostics

A0400 [W] Le code diagnostique est incompatible quant au sexe

DIA[i].inIndex && strlen(DIA[i].Data[ICD_SEX]) > 0 && strcmp(DIA[i].Data[ICD_SEX],MB->V1101) != 0

A0410 [W] Incompatibilité du code CIM quant à l'âge pour : a) Nourrissons (il s'agit ici de nouveau-nés principalement. Toutefois, des patients dont l'âge varie entre 1 et 364/5 jours figurent également dans ce groupe)

DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'a') == 0 && MB->AGED > 365

A0411 [W] Incompatibilité du code CIM quant à l'âge pour : b) Enfants en bas-âge (âge variant entre 1 jour et 2 ans)

DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'b') == 0 && (MB->AGEJ >= 2 || MB->AGED == 0)

A0412 [W] Incompatibilité du code CIM quant à l'âge pour : c) Enfant (âge variant entre 1 jour et 10 ans)

DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'c') == 0 && (MB->AGEJ >= 10 || MB->AGED == 0)

A0413 [W] Incompatibilité du code CIM quant à l'âge pour : d) Enfants et adolescents (âge variant entre 1 jour et 19 ans)

DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'd') == 0 && (MB->AGEJ >= 19 || MB->AGED == 0)

A0414 [W] Incompatibilité du code CIM quant à l'âge pour : e) Bébé âgé de plus de 27 jours (période post-néonatale)

DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'e') == 0 && MB->AGED < 28

A0415 [W] Incompatibilité du code CIM quant à l'âge pour : f) Age supérieur à celui d'un nourrisson (c'est-à-dire âgé d'un an ou plus)

DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'f') == 0 && MB->AGEJ < 1

A0416 [W] Incompatibilité du code CIM quant à l'âge pour : g) Puberté (âge variant entre 8 et 19 ans)

DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'g') == 0 && (MB->AGEJ < 8 || MB->AGEJ > 19)

A0417 [W] Incompatibilité du code CIM quant à l'âge pour : h) Patientes en âge de procréer (entre 12 et 55 ans)

DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'h') == 0 && (MB->AGEJ < 12 || MB->AGEJ > 55)

A0418 [W] Incompatibilité du code CIM quant à l'âge pour : j) Adultes (à partir de 15 ans)

DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'j') == 0 && MB->AGEJ < 15

A0419 [W] Incompatibilité du code CIM quant à l'âge pour : k) Adultes (à partir de 30 ans)

DIA[i].inIndex && strcmp(DIA[i].Data[ICD_AGEGRP], 'k') == 0 && MB->AGEJ < 30

A05. Attributs des diagnostics

A0510 [W] Ce code CIM exige une latéralité.

i != 1 && DIA[i].inIndex && strcmp(DIA[i].Data[ICD_LATERAL], '1') == 0

A09. Tests formels

A0910 [H] Malformaion congénitale probable.

`strlen(DIA[i].Key) > 0 && strcmp(DIA[i].Key,'Q',1) == 0 && strcmp(MB->V0301,'1') == 0`

A0920 [H] Diagnostic à signaler obligatoirement

`DIA[i].inIndex && (hasKeyLikeLen(DIA[i].Key,4,'|A392|A393|A394|A395|A398|A399|A403|A413|A481|A482|A491|A492|A810|A841|B160|B161|'| hasKeyLikeLen(DIA[i].Key,3,'|J09|J13|J14|'))`

A0990 [X] Ce diagnostic a déjà été utilisé plus haut (doublon)

`strlen(DIA[i].Key) > 0 && strcmp(DIA[j].Key, DIA[i].Key) == 0`

A1. Traitements

A10. Traitement principal

A1000 [W] Traitement principal : La donnée manque

`BEH[0].isNull && isn(MB->V0202,2,_AC) && !isn(MB->V1401,7,_7_HKSTH) && strcmp(DIA[0].Key,'Z38',3) != 0 && strcmp(DIA[0].Key,'Z758',4) != 0`

A1001 [H] Traitement principal : La donnée manque

`BEH[0].isNull && isn(MB->V0202,2,_AC) && isn(MB->V1401,7,_7_HKSTH) && strcmp(MB->V1401,'M200') != 0 && strcmp(DIA[0].Key,'Z38',3) != 0 && strcmp(DIA[0].Key,'Z758',4) != 0`

A1005 [E] Les codes supplémentaires ne sont pas des codes de traitements principaux

`*BEH[0].Data[CHOP_VALIDHB] == '2'`

A12. Validité des codes CHOP

A1200 [E] Le code du traitement (CHOP) est invalide

`!BEH[i].isNull && !BEH[i].inIndex`

A1201 [E] Le code CHOP n'était plus valable au moment de la sortie

`BEH[i].inIndex && strlen(BEH[i].Data[CHOP_VALID_TO]) > 0 && MB->dateV1501.hasYear && MB->dateV1501.Year > atoi(BEH[i].Data[CHOP_VALID_TO])`

A1202 [E] Le code CHOP n'était pas encore valable au moment de la sortie

`BEH[i].inIndex && strlen(BEH[i].Data[CHOP_VALID_FROM]) > 0 && MB->dateV1501.hasYear && MB->dateV1501.Year < atoi(BEH[i].Data[CHOP_VALID_FROM])`

A1210 [E] Procédure (CHOP) non terminale

`BEH[i].inIndex && StatYear > 2009 && StatYear >= atoi(BEH[i].Data[CHOP_VALID_FROM]) && StatYear <= atoi(BEH[i].Data[CHOP_VALID_TO]) && (StatYear < atoi(BEH[i].Data[CHOP_TERM_FROM]) || StatYear > atoi(BEH[i].Data[CHOP_TERM_TO]))`

A1222 [E] Le code de traitement à deux caractères n'est pas spécifique (CHOP)

`!BEH[i].inIndex && strcmp(BEH[i].Data[CHOP_TYP],'2') == 0`

A1223 [E] Le code de traitement à trois caractères n'est pas spécifique (CHOP)

```
BEH[i].inIndex && strcmp(BEH[i].Data[CHOP_TYP], '3') == 0
```

A13. Regles de coage

A1310 [W] Le code du traitement n'est possible que pour des cas de décès

```
strcmp(MB->V1502, '5') != 0 && strcmp(BEH[i].Key, '8981', 4) == 0
```

A14. Plausibilité des codes CHOP

A1400 [W] Le code du traitement est incompatible avec le sexe

```
BEH[i].inIndex && strlen(BEH[i].Data[CHOP_SEX]) > 0 && strcmp(BEH[i].Data[CHOP_SEX], MB->V1101) != 0 && !hasF64
```

A1401 [E] Le code du traitement du chapitre 13 est incompatible avec le sexe (masculin)

```
strcmp(MB->V0202, 'A') == 0 && BEH[i].inIndex && strcmp(MB->V1101, '1') == 0 && strcmp(BEH[i].Key, '72', 2) >= 0 && strcmp(BEH[i].Key, '75', 2) <= 0
```

A1410 [W] Le code diagnostique est incompatible avec l'âge (il n'est utilisable que pour la période péri-/néonatale)

```
DIA[i].inIndex && DIA[i].Key[0] == 'P' && MB->dateV1201.DateValue - MB->dateV1102.DateValue > 28 && (DIA[i].Key[1] == '1' || isSubKeyIn(DIA[i].Key, 1, 2, 16, _16_DIA_PERINAT_LEN2) || isSubKeyIn(DIA[i].Key, 1, 3, 10, _10_DIA_PERINAT_LEN3))
```

A1412 [W] Le code du traitement du chapitre 13 est incompatible avec l'âge (< 12 ans)

```
strcmp(MB->V0202, 'A') == 0 && BEH[i].inIndex && atoi(MB->V1103) < 12 && strcmp(BEH[i].Key, '72', 2) >= 0 && strcmp(BEH[i].Key, '75', 2) <= 0
```

A1417 [W] Incompatibilité du code CHOP quant à l'âge pour : i) Patientes en âge de procréer (entre 12 et 55 ans)

```
BEH[i].inIndex && strcmp(BEH[i].Data[CHOP_AGEGRP], 'i') == 0 && (MB->AGEJ < 12 || MB->AGEJ > 55)
```

A18. Début du traitement

A1800 [E] Début du traitement : La donnée manque

```
!BEH[i].isNull && BEH[i].dateDatum.isNull
```

A1810 [E] Début du traitement : La donnée concernant la date est invalide

```
!BEH[i].dateDatum.isNull && !BEH[i].dateDatum.hasDate
```

A1811 [E] Début du traitement : La donnée concernant l'heure est invalide

```
BEH[i].dateDatum.hasDate && !BEH[i].dateDatum.hasHour && (BEH[i].dateDatum.Length > 8 || i == 0)
```

A1815 [E] Les traitements supplémentaires sont transmis sans indication d'heure

```
i > 0 && BEH[i].dateDatum.hasDate && strlen(BEH[i].dateDatum.Value) > 8
```

A1821 [E] MD : La date d'admission doit précéder la date du traitement

```
strcmp(MB->V0202,'A') == 0 && (MB->dateV1201.hasDate && BEH[i].dateDatum.hasDate && MB->dateV1201.DateValue > BEH[i].dateDatum.DateValue) || (MB->dateV1201.hasHour && BEH[i].dateDatum.hasHour && MB->dateV1201.DateValue == BEH[i].dateDatum.DateValue && MB->dateV1201.Hour > BEH[i].dateDatum.Hour)
```

A1822 [E] MD : La date du traitement doit précéder la date de sortie

```
(MB->dateV1501.hasDate && BEH[i].dateDatum.hasDate && MB->dateV1501.DateValue < BEH[i].dateDatum.DateValue) || (MB->dateV1501.hasHour && BEH[i].dateDatum.hasHour && MB->dateV1501.DateValue == BEH[i].dateDatum.DateValue && MB->dateV1501.Hour < BEH[i].dateDatum.Hour)
```

A1825 [E] La date de naissance ne peut être postérieure à celle du début du traitement principal

```
dateV1102.hasYear && dateV1702.hasDate && dateV1102.DateValue > dateV1702.DateValue
```

A19. Tests formels

A1910 [X] La localisation des stents doit être codé avant le code 00.4A.xx

```
BEH[i].inIndex && strcmp(BEH[i].Key,'004A',4) == 0 && (i == 0 || !BEH[i-1].inIndex || BEH[i-1].KeyLen != 6 || strstr('|397211|397221|397921|397922|397923|397924|397925|397926|397927|397929|399931|399932|',BEH[i-1].Key) != NULL)
```

A1911 [X] La localisation des stents doit être codé le même jour que le code 00.4A.xx

```
BEH[i].inIndex && strcmp(BEH[i].Key,'004A',4) == 0 && i > 0 && BEH[i-1].inIndex && BEH[i-1].KeyLen == 6 && strstr('|397211|397221|397928|397921|397922|397924|397925|397926|397927|397929|399931|399932|',BEH[i-1].Key) != NULL && BEH[i].dateDatum.DateValue != BEH[i-1].dateDatum.DateValue
```

A1920 [X] Code imprécis - ne doit pas être utilisé

```
strcmp(BEH[i].Key, '399530') == 0 || strcmp(BEH[i].Key, '399540') == 0 || strcmp(BEH[i].Key, '399570') == 0 || strcmp(BEH[i].Key, '399580') == 0 || strcmp(BEH[i].Key, '3995A0') == 0 || strcmp(BEH[i].Key, '3995C0') == 0 || strcmp(BEH[i].Key, '3995D0') == 0 || strcmp(BEH[i].Key, '3995E0') == 0
```

A1990 [X] Ce traitement a déjà été utilisé plus haut (doubleton)

```
strlen(BEH[i].Key) > 0 && strcmp(BEH[j].Key, BEH[i].Key) == 0
```

B. Données administratives

B0. Données du patient

B0100 [E] Sexe : La donnée manque

```
strlen(V1101) == 0
```

B0101 [E] Sexe : La donnée est invalide

```
strlen(V1101) > 0 && !isIn(V1101,2,_12)
```

B0110 [E] Date de naissance : La donnée manque

```
dateV1102.isNull
```

B0111 [E] Date de naissance : La donnée est invalide

```
!dateV1102.isNull && !(dateV1102.isYYYYMMDD || dateV1102.isYYYY0000)
```

B0115 [E] La date de naissance doit précéder la date d'admission

```
dateV1102.hasYear && dateV1201.hasDate && dateV1102.DateValue > dateV1201.DateValue
```

B0120 [E] Age lors de l'admission : La donnée manque

```
strlen(V1103) == 0
```

B0121 [E] Age lors de l'admission : En dehors du domaine valable

```
strlen(V1103) > 0 && (atol(V1103) < 0 || atol(V1103) > 120)
```

B0122 [E] Naissance (enfant)/indication concernant l'âge : Un nouveau-né a l'âge 0

```
strcmp(V1203,'3') == 0 && atol(V1103) != 0
```

B0130 [E] Domicile : La donnée manque

```
strlen(V1104) == 0
```

```
ACTION : V1104="XXXX";
```

B0131 [E] Domicile : La donnée est invalide

```
!(V1104._regio.inIndex || V1104._plz.inIndex || V1104._wregio.inIndex || V1104._lregio.inIndex || V1104._natio.inIndex)
```

B0132 [W] Domicile : Lorsque le domicile se trouve en Suisse, veuillez indiquer le code de la région (ou NPA)

```
strcmp(V1104,'CHE') == 0
```

B0140 [E] Nationalité : La donnée manque

`strlen(V1105) == 0`

`ACTION : V1105="XXX" ;`

B0141 [E] Nationalité : La donnée est invalide

`!V1105_natio.isNull && !(V1105_natio.inIndex || V1105_wregio.inIndex)`

B0200 [E] Année de la naissance : Se situe plus de 120 ans en arrière

`dateV1102.hasYear && dateV1102.Year < MX->dateV0004.Year - 120`

B0201 [E] Date de naissance / date d'admission : Est en contradiction avec l'âge indiqué

`dateV1201.hasYear && dateV1102.hasYear && strlen(V1103) > 0 && !(dateV1201.Year - dateV1102.Year == atoi(V1103) || dateV1201.Year - dateV1102.Year == atoi(V1103)+1)`

B0202 [E] Date de naissance de l'enfant : La date de naissance complète doit être introduite pour les enfants jusqu'à deux ans

`isInt(V1103) && atoi(V1103) < 2 && dateV1102.isYYYY0000`

B0203 [E] Date de naissance du décédé : En cas de décès, la date de naissance complète doit être introduite

`strcmp(V1502,'5') == 0 && dateV1102.isYYYY0000`

B0300 [W] Durée de la grossesse_12M : La donnée manque

`MB->AGED < 365 && strlen(V4816) == 0 && !MB->hasMN`

B0301 [W] Durée de la grossesse_12M : Le nombre de semaines est improbable

`strlen(V4816) == 3 && ((atoi(V4816)/10) < 16 || (atoi(V4816)/10) > 44)`

B0302 [E] Durée de la grossesse_12M : Le nombre de jours est invalide

`strlen(V4816) == 3 && strchr('0123456',V4816[2]) == NULL`

B0305 [W] Durée de la grossesse_12M : format invalide

`strlen(V4816) > 0 && ((!isHex(V4816) && strlen(V4816) == 16) || (strlen(V4816) != 3 && strlen(V4816) != 16))`

B1. Séjour hospitalier

B10. Admission

B1000 [E] Date lors de l'admission : La donnée manque

`dateV1201.isNull`

B1001 [E] Date lors de l'admission : La donnée concernant la date est invalide

`!dateV1201.isNull && !dateV1201.hasDate`

ACTION : V1201="" ;

B1002 [E] Heure lors de l'admission : La donnée concernant l'heure est invalide

`dateV1201.Length>8 && dateV1201.hasDate && !dateV1201.hasHour`

B1005 [E] Admission aux urgences : La date de l'admission doit inclure l'heure

`V1203[0] == '1' && dateV1201.hasDate && !dateV1201.hasHour`

B1010 [E] Séjour avant l'admission : La donnée manque

`strlen(V1202) == 0`

ACTION : V1202="9" ;

B1011 [E] Séjour avant l'admission : La donnée est invalide

`strlen(V1202) > 0 && !isIn(V1202,9,_9_AUFVOR)`

ACTION : V1202="9" ;

B1020 [E] Mode d'admission : La donnée manque

`strlen(V1203) == 0`

ACTION : V1203="9" ;

B1021 [E] Mode d'admission : La donnée est invalide

`strlen(V1203) > 0 && !isIn(V1203,7,_7_1234589)`

ACTION : V1203="9" ;

B1022 [E] Une mère ne doit jamais avoir Naissance comme mode d'admission

`strlen(MB->V1103) > 0 && atoi(MB->V1103)>0 && strcmp(MB->V1203,'3') == 0`

B1023 [W] Date d'admission/de naissance, mode d'admission : Signaler une naissance dans le mode d'admission

`strcmp(MB->V1103,'0') == 0 && MB->dateV1102.DateValue == MB->dateV1201.DateValue && strcmp(MB->V1203,'3') != 0 && !hasZ38_147`

B1030 [E] Instance décidant l'envoi : La donnée manque

`strlen(V1204) == 0`

ACTION : V1204="9" ;

B1031 [E] Instance décidant l'envoi : La donnée est invalide

`strlen(V1204) > 0 && !isIn(V1204,8,_8_INSTANZ)`

ACTION : V1204="9" ;

B1055 [H] Date de l'admission / début de la période de relevé : Se situe plus de 2 ans avant le début de la période de relevé

```
dateV1201.hasDate && MX->dateV0006.hasDate && MX->dateV0006.Year - dateV1201.Year > 2
```

B1060 [E] MB : La date d'admission doit précéder la date du traitement

```
strcmp(MB->V0202,'A') == 0 && (dateV1201.hasDate && dateV1702.hasDate && dateV1201.DateValue > dateV1702.DateValue) || (dateV1201.hasHour && dateV1702.hasHour && dateV1201.DateValue == dateV1702.DateValue && dateV1201.Hour > dateV1702.Hour)
```

B1061 [E] L'heure de l'admission doit précéder l'heure du traitement principal

```
dateV1201.hasHour && dateV1702.hasHour && dateV1201.DateValue == dateV1702.DateValue && dateV1201.Hour > dateV1702.Hour
```

B1065 [E] La date d'admission doit précéder la date de sortie

```
dateV1201.hasDate && dateV1501.hasDate && dateV1201.DateValue > dateV1501.DateValue
```

B1066 [E] L'heure d'admission doit précéder l'heure de sortie

```
dateV1201.hasHour && dateV1501.hasHour && dateV1201.DateValue == dateV1501.DateValue && dateV1201.Hour > dateV1501.Hour
```

B11. Séjour

B1110 [E] Type de traitement : La donnée manque

```
strlen(V1301) == 0
```

```
ACTION : V1301="9";
```

B1111 [E] Type de traitement : La donnée est invalide

```
strlen(V1301) > 0 && !isIn(V1301,4,_4_BEHART)
```

```
ACTION : V1301="9";
```

B1112 [W] Type de traitement : La donnée n'est pas plausible

```
strcmp(V1301,'9') == 0
```

B1113 [E] Type de traitement : semi-hospitalier n'est plus valable à partir de 2009

```
strlen(V1301) == 1 && V1301[0] == '2' && dateV1501.Year > 2008 && !(strcmp(V1401,'M500') == 0 && strcmp(V0104,'BE') == 0)
```

```
ACTION : doWriteRec=FALSE;
```

B1115 [H] Durée du séjour non valable pour traitement ambulatoire

```
strcmp(V1301,'1') == 0 && dateV1201.hasDate && dateV1501.hasDate && dateV1501.DateValue - dateV1201.DateValue > 1
```

B1120 [E] Classe : La donnée manque

`strlen(V1302) == 0`

ACTION : V1302="9";

B1121 [E] Classe : La donnée est invalide

`strlen(V1302) > 0 && !isIn(V1302,4,_4_KLASSE)`

ACTION : V1302="9";

B1122 [H] Classe : La donnée n'est pas plausible

`strcmp(V1301,'3') == 0 && strcmp(V1302,'9') == 0`

B1130 [E] Centre de prise en charge des coûts : La donnée manque

`strlen(V1401) == 0`

B1131 [E] Centre de prise en charge des coûts : La donnée est invalide

`strlen(V1401) > 0 && !((StatYear >= 2016 && isIn(V1401,16,_16_KST)) || (StatYear < 2016 && isIn(V1401,14,_14_KST)))`

B1135 [W] Centre de charge / âge : Limite d'âge supérieure dépassée pour la pédiatrie (M400)

`strlen(V1103) > 0 && strcmp(V1401,'M400') == 0 && atol(V1103) > 18`

B1136 [W] Centre de charge / âge : Limite d'âge inférieure pas atteinte pour la gériatrie (M900)

`strlen(V1103) > 0 && strcmp(V1401,'M900') == 0 && atol(V1103) < 30`

B1140 [E] Prise en charge des soins de base : La donnée manque

`strlen(V1402) == 0`

ACTION : V1402="9";

B1141 [E] Prise en charge des soins de base : La donnée est invalide

`strlen(V1402) > 0 && !isIn(V1402,7,_1234589)`

ACTION : V1402="9";

B1150 [W] Vacances, congés administratifs : La donnée manque

`strlen(V1304) == 0`

B1151 [E] Vacances, congés administratifs : La donnée est invalide

`strlen(V1304) > 0 && (atol(V1304) < 0 || atol(V1304) > 9999)`

ACTION : V1304="";

B1152 [E] Vacances, congés administratifs : La donnée dépasse la durée du séjour total

`dateV1501.hasDate && dateV1201.hasDate && atoi(V1304) > 0 && (dateV1501.DateValue > dateV1201.DateValue) && atoi(V1304) > (dateV1501.DateValue - dateV1201.DateValue + 1) * 24`

B12. Sortie

B1200 [E] Date de sortie : La donnée manque (cas statistique A)

`strlen(V1501) == 0 && V0202[0] == 'A'`

B1201 [E] Date de sortie : La donnée concernant la date est invalide

`!dateV1501.isNull && !dateV1501.hasDate`

B1202 [E] Heure de sortie : La donnée concernant l'heure est invalide

`dateV1501.Length > 8 && dateV1501.hasDate && !dateV1501.hasHour`

B1210 [E] Décision de sortie : La donnée manque

`strlen(V1502) == 0 && strcmp(V0202,'A') == 0`

`ACTION : V1502="9";`

B1211 [E] Décision de sortie : La donnée est invalide

`strlen(V1502) > 0 && !isIn(V1502,7, _1234589)`

`ACTION : V1502="9";`

B1220 [E] Séjour après la sortie : La donnée manque

`strlen(V1503) == 0 && V0202[0] == 'A'`

`ACTION : V1503="9";`

B1221 [E] Séjour après la sortie : La donnée est invalide

`strlen(V1503) > 0 && !isIn(V1503,10, _0123456789)`

`ACTION : V1503="9";`

B1230 [E] Prise en charge après la sortie : La donnée manque

`strlen(V1504) == 0 && strcmp(V0202,'A') == 0`

`ACTION : V1504="9";`

B1231 [E] Prise en charge après la sortie : La donnée est invalide

`strlen(V1504) > 0 && !isIn(V1504,8, _01234589)`

`ACTION : V1504="9";`

B1250 [W] Décès : Après le décès, le séjour et la prise en charge doivent tous deux avoir la valeur 0

`strcmp(V1502,'5') == 0 && (strcmp(V1503,'0') != 0 || strcmp(V1504,'0') != 0)`

B1251 [E] Date de sortie en cas de décès : La donnée doit comprendre l'heure

```
strcmp(V1502,'5') == 0 && !dateV1501.hasHour
```

B1252 [E] Séjour/prise en charge après la sortie : Cette entrée n'est possible que dans des cas de décès

```
(strcmp(V1503,'0') == 0 || strcmp(V1504,'0') == 0) && strcmp(V1502,'5') != 0
```

B1300 [W] Statut DRG : Veuillez saisir le statut

```
strcmp(MB->V0202,'A') == 0 && strlen(V4801) == 0
```

B1301 [E] Statut DRG : La donnée est invalide

```
strlen(V4801) > 0 && !isIn(V4801,2,_01)
```

B1310 [W] Un patient en attente a toujours un status DRG de 0

```
strcmp(DIA[0].Key, 'Z758') == 0 && *(MB->V1203) == '4' && *MB->MD->V4801 == '1'
```

B2. Soins intensifs

B2000 [W] Séjour en soins intensifs : La donnée manque

```
strcmp(MB->V0202,'A') == 0 && strlen(V1303) == 0
```

B2001 [E] Séjour en soins intensifs : La donnée est invalide

```
strlen(V1303) > 0 && (atol(V1303) < 0 || atol(V1303) > 9999)
```

```
ACTION : V1303="9999" ;
```

B2010 [W] Un séjour en soins intensifs est déclaré dans les données minimales. Une partie des champs «Médecine intensive MD» doit être rempli.

```
strlen(MB->V1303) > 0 && atol(MB->V1303) != 0 && (strlen(V4401) == 0 || strlen(V4402) == 0) && strcmp(MB->V0202, 'A') == 0
```

B2011 [E] Ce cas comprend des heures en soins intensifs, mais le champ NEMS est vide

```
strlen(MB->V1303) > 0 && atol(MB->V1303) != 0 && (strlen(V4404) == 0 || !(atol(V4404) > 0))
```

B2110 [E] La durée de la ventilation artificielle ne peut pas être plus courte que 0 ou plus longue que 9999 heures

```
strlen(V4401) > 0 && (atol(V4401) < 0 || atol(V4401) > 99999)
```

B2120 [H] La durée de la ventilation artificielle ne peut être plus longue que le nombre d'heures IPS

```
strcmp(MB->V0202,'A') == 0 && strlen(V4401) > 0 && (atol(V4401) > (atol(MB->V1303)+15)) && atol(MB->V1303) > 0
```

B2130 [E] La durée de la ventilation artificielle ne peut pas être plus longue que le séjour à l'hôpital

```
strcmp(MB->V0202,'A') == 0 && strlen(V4401) > 0 && atol(V4401) > (MB->STAYH+15)
```

B2210 [E] La gravité de la maladie aiguë ne peut être plus petite que 0 ou plus grande que 999 points.

`strlen(V4402) > 0 && (atol(V4402) < 0 || atol(V4402) > 999)`

B2220 [E] Lorsque la gravité a été entrée, le genre de score doit également être introduit.

`strlen(V4402) > 0 && strlen(V4403) == 0`

B2221 [E] Le genre de score n'est pas valable

`strlen(V4403) > 0 && !isin(V4403,3,_3_SCOREART)`

B2225 [E] Lorsque le genre de score a été entré, la gravité doit également être introduite.

`strlen(V4403) > 0 && strlen(V4402) == 0`

B2310 [E] NEMS : Le total de toutes les séances ne peut être plus petit que 0 ou plus grand que 999999

`strlen(V4404) > 0 && (atol(V4404) < 0 || atol(V4404) > 999999)`

B2400 [E] Séjour en soins intensifs : La donnée dépasse la durée du séjour total

`dateV1501.hasDate && dateV1201.hasDate && atol(V1303) > 0 && (dateV1501.DateValue > dateV1201.DateValue) && atol(V1303) > (dateV1501.DateValue - dateV1201.DateValue + 1) * 24`

B3. Données sur les nouveau-nés

B30. Données de la mère

B3000 [E] Date de naissance de la mère : La donnée manque

`strlen(V2301) == 0`

B3001 [E] Date de naissance de la mère : La donnée concernant la date est invalide (AAAAMMJJ)

`strlen(V2301) > 6 && dateV2301.hasMonth && (strlen(V2301) != 8 || (dateV2301.Day > 0 && !dateV2301.isYYYYMMDD))`

B3002 [E] Date de naissance de la mère : La donnée concernant l'année est invalide

`!dateV2301.isNull && (!dateV2301.hasYear || dateV2301.Year < 1900)`

B3003 [E] Date de naissance de la mère : La donnée concernant le mois manque/est erronée

`!dateV2301.isNull && !dateV2301.hasMonth`

B3005 [E] Date de naissance de la mère : ne doit pas équivaloir à celle de l'enfant

`strlen(V2301) > 0 && strcmp(V2301,MB->V1102,6) == 0`

B3010 [W] Durée de la grossesse 1 : La donnée manque

`strlen(V2302) == 0`

B3011 [E] Durée de la grossesse 1 : Le nombre de jours est invalide

`strlen(V2302) > 0 && (strlen(V2302) != 3 || strchr('0123456',V2302[2]) == NULL)`

B3012 [E] Durée de la grossesse 1 : Le nombre de semaines n'est pas valable

`strlen(V2302) > 0 && ((atol(V2302)/10) < 16 || (atol(V2302)/10) > 44)`

B3015 [W] Durée de la grossesse 1 : Le nombre de semaines est improbable

`strlen(V2302) > 0 && ((atol(V2302)/10) < 16 || (atol(V2302)/10) > 44)`

B3020 [W] Durée de la grossesse 2 : La donnée manque

`strlen(V2303) == 0`

B3021 [E] Durée de la grossesse 2 : Le nombre de jours est invalide

`strlen(V2303) > 0 && (strlen(V2303) != 3 || strchr('0123456',V2303[2]) == NULL)`

B3022 [E] Durée de la grossesse 2 : Le nombre de semaines n'est pas valable

`strlen(V2303) > 0 && ((atol(V2303) / 10) < 16 || (atol(V2303) / 10) > 45)`

B3025 [W] Durée de la grossesse 2 : Le nombre de semaines est improbable

`strlen(V2303) > 0 && ((atol(V2303) / 10) < 16 || (atol(V2303) / 10) > 45)`

B3026 [W] Durée de la grossesse 2 : Elle est de 4 semaines plus longue que celle de la grossesse 1

`strlen(V2302) > 0 && strlen(V2303) > 0 && ((atol(V2303) / 10) - (atol(V2302) / 10) > 4)`

B3040 [H] Total de grossesses précédentes : La donnée manque

`strlen(V2304) == 0`

B3041 [E] Total de grossesses précédentes : En dehors du domaine valable

`strlen(V2304) > 0 && (atol(V2304) < 0 || atol(V2304) > 40)`

B3050 [H] Nombre de naissances vivantes précédentes : La donnée manque

`strlen(V2305) == 0`

B3051 [E] Nombre de naissances vivantes précédentes : En dehors du domaine valable

`strlen(V2305) > 0 && (atol(V2305) < 0 || atol(V2305) > 20)`

B3060 [H] Nombre de fausses couches et de mortinaissances précédentes : La donnée manque

`strlen(V2306) == 0`

B3061 [E] Nombre de fausses couches et de mortinaissances précédentes : En dehors du domaine valable

`strlen(V2306) > 0 && (atol(V2306) < 0 || atol(V2306) > 20)`

B3065 [W] Données supplémentaires nouveau-nés : Données contradictoires concernant les grossesses

`strlen(V2305) > 0 && strlen(V2306) > 0 && strlen(V2307) > 0 && atol(V2304) > (atol(V2305) + atol(V2306) + atol(V2307))`

B3070 [H] Nombre d'interruptions de grossesses précédentes : La donnée manque

`strlen(V2307) == 0`

B3071 [E] Nombre d'interruptions de grossesses précédentes : En dehors du domaine valable

`strlen(V2307) > 0 && (atol(V2307) < 0 || atol(V2307) > 20)`

B3080 [E] Transfert de la mère dans un autre hôpital : La donnée manque

`strlen(V2308) == 0`

B3081 [E] Transfert de la mère dans un autre hôpital : La donnée est invalide

`strlen(V2308) > 0 && !isIn(V2308,2,_01)`

`ACTION : V2308="" ;`

B31. Données de l'enfant

B3100 [E] Données des nouveau-nés : L'enregistrement supplémentaire manque

`strcmp(V0202,'A') == 0 && strcmp(V0301,'1') == 0 && !hasMN`

`ACTION : V0301="0" ;`

B3101 [E] Un enfant né dans l'établissement doit toujours être indiqué dans l'enregistrement supplémentaire MN

`*MB->V0202 == 'A' && *MB->V1203 == '3' && *MB->V1103 == '0' && !MB->hasMN`

B3102 [H] Données supplémentaires nouveau-nés / Centre de charge : En principe, les nouveau-nés sont en obstétrique/gynécologie ou en pédiatrie (M300, M400)

`strcmp(V2101,'MN') == 0 && !isIn(MB->V1401,2,_2_KST_MN)`

B3104 [E] Données des nouveau-nés / Age lors de l'admission : L'âge des nouveau-nés ne peut être que 0

`strcmp(V2101,'MN') == 0 && atol(MB->V1103) != 0`

B3105 [E] Etat à la naissance / Mode de sortie : Données contradictoires

`strcmp(V2201,'0') == 0 && strcmp(MB->V1502,'5') != 0`

B3110 [H] Numéro de naissance interne : La donnée manque

`strlen(V2102) == 0`

`ACTION : V2102="0" ;`

B3111 [E] Numéro de naissance interne : En dehors du domaine valable

`strlen(V2102) > 0 && atoi(V2102) > 9999`

`ACTION : V2102="0";`

B3130 [E] Etat à la naissance : La donnée manque

`strlen(V2201) == 0`

B3131 [E] Etat à la naissance : La donnée est invalide

`strlen(V2201) > 0 && !isIn(V2201,2,_01)`

B3140 [E] Naissances multiples : La donnée manque

`strlen(V2202) == 0`

B3141 [E] Naissances multiples : La donnée est invalide

`strlen(V2202) > 0 && !isIn(V2202,9,_9_MEHRLING)`

B3150 [E] Rang lors de naissances multiples : La donnée manque / est incorrecte pour des naissances multiples

`atoi(V2202) > 1 && (strlen(V2203) == 0 || atoi(V2203) > atoi(V2202))`

B3151 [H] Les naissances uniques doivent avoir un rang égal à 1 à la naissance

`strcmp(V2202,'1') == 0 && strlen(V2203) > 0 && strcmp(V2203,'1') != 0`

`ACTION : V2203="1";`

B3152 [H] Rang lors de naissances multiples : La donnée est invalide

`strlen(V2203) > 0 && !isIn(V2203,9,_9_GEBRANG)`

`ACTION : V2203="1";`

B3160 [E] Poids à la naissance (gr.) : La donnée manque

`strlen(V2204) == 0`

B3161 [E] Poids à la naissance (gr.) : En dehors du domaine valable

`strlen(V2204) > 0 && (atoi(V2204) < 0 || atoi(V2204) > 7000)`

B3162 [W] Poids à la naissance (gr.) : La donnée n'est pas plausible pour un bébé né vivant

`strlen(V2204) > 0 && strcmp(V2201,'1') == 0 && atoi(V2204) <= 500`

B3163 [W] Poids à la naissance (gr.)/Durée de la grossesse 1 : La donnée n'est pas plausible pour un bébé mort-né

`strlen(V2204) > 0 && strcmp(V2201,'0') == 0 && (((atoi(V2302) <= 217) && (atoi(V2204) > 500)) || ((atoi(V2302) > 217) && (atoi(V2204) <= 500)))`

B3164 [W] Poids à la naissance (gr.) : La donnée n'est pas plausible

`strlen(V2204) > 0 && atol(V2204) <= 7000 && atol(V2204) >= 6000`

B3165 [H] Poids à l'admission : La donnée manque.

`strlen(V4501) == 0 && strcmp(MB->V1203,'3') == 0`

B3166 [E] Le poids à l'admission doit être relevé pour tout nourrisson âgé d'au plus 28 jours.

`strcmp(MB->V0202,'A') == 0 && MB->AGED < 28 && strlen(V4501) == 0`

B3167 [W] Poids à l'admission d'un nourrisson âgé d'au plus 12 mois : La donnée manque.

`strcmp(MB->V0202,'A') == 0 && strlen(V4501) == 0 && strcmp(MB->V1203,'3') != 0 && MB->AGEJ < 1`

B3168 [E] Le poids à l'admission ne peut être plus petit que 0 ou plus grand que 99999 grammes.

`strlen(V4501) > 0 && (atol(V4501) < 0 || atol(V4501) > 16000)`

B3169 [E] Lorsque le mode d'entrée est = 3 Naissance, le poids d'un nourrisson à l'admission doit correspondre au poids à la naissance figurant dans le supplément nouveau-nés.

`MB->hasMN && strlen(V4501) > 0 && strcmp(MB->V1203,'3') == 0 && atol(V4501) != atol(MN->V2204)`

B3170 [E] Longueur du corps (cm) : La donnée manque

`strlen(V2205) == 0`

B3171 [E] Longueur du corps (cm) : En dehors du domaine valable

`strlen(V2205) > 0 && strcmp(V2201,'1') == 0 && (strcmp(V2205,'0') == 0 || atol(V2205) > 65)`

B3172 [W] Longueur du corps (cm) : valeur improbable

`strlen(V2205) > 0 && strcmp(V2201,'1') == 0 && (atol(V2205) < 25 && atol(V2205) > 0)`

B3174 [W] Relation entre poids à la naissance et longueur du corps : En dehors du domaine valable

`atol(V2205) != 0 && (atol(V2204) / atol(V2205) < 20 || atol(V2204) / atol(V2205) > 120)`

B3175 [W] Tour de tête à la naissance : La donnée manque.

`strcmp(MB->V0202,'A') == 0 && strlen(V4502) == 0 && MB->hasMN`

B3176 [H] Tour de tête à la naissance : cette données n'est saisie que pour les nouveau-nés

`strcmp(MB->V0202,'A') == 0 && strlen(V4502) > 0 && !MB->hasMN`

`ACTION : V4502="" ;`

B3177 [W] Le tour de tête à la naissance ne peut pas être plus petit que 15 ou plus grand que 50 cm.

`strlen(V4502) > 0 && strcmp(MB->V1502,'5') == 1 && (atol(V4502) < 15 || atol(V4502) > 50)`

B3180 [E] Malformations congénitales : La donnée manque

`strlen(V2206) == 0`

B3181 [E] Malformations congénitales : La donnée est invalide

`strlen(V2206) > 0 && !isIn(V2206,3,_3_KONGEN_MISSB)`

B3190 [E] Transfert de l'enfant dans un autre hôpital : La donnée manque

`strlen(V2207) == 0`

B3191 [E] Transfert de l'enfant dans un autre hôpital : La donnée est invalide

`strlen(V2207) > 0 && !isIn(V2207,2,_01)`

B3195 [E] Heure de la naissance : La donnée manque

`strlen(V2103) == 0`

B3196 [E] Heure de la naissance : La donnée est invalide

`strlen(V2103) != 4 && (atoi(V2103)/100 > 23 || atoi(V2103)%100 > 59)`

B4. Données psychiatriques

B40. Patient

B4000 [W] Etat civil : La donnée manque

`strlen(V3201) == 0`

`ACTION : V3201="9";`

B4001 [E] Etat civil : La donnée est invalide

`strlen(V3201) > 0 && !isIn(V3201,6,_6_ZIVILSTAND)`

`ACTION : V3201="9";`

B4010 [W] Occupation avant l'admission (temps partiel) : La donnée manque

`strlen(V3203) == 0`

B4011 [E] Occupation avant l'admission (temps partiel) : La donnée est invalide

`strlen(V3203) > 0 && !isIn(V3203,2,_01)`

B4012 [W] Occupation avant l'admission (temps complet) : La donnée manque

`strlen(V3204) == 0`

B4013 [E] Occupation avant l'admission (temps complet) : La donnée est invalide

`strlen(V3204) > 0 && !isIn(V3204,2,_01)`

B4014 [W] Occupation avant l'admission (sans activité ou au chômage) : La donnée manque

`strlen(V3205) == 0`

B4015 [E] Occupation avant l'admission (sans activité ou au chômage) : La donnée est invalide

`strlen(V3205) > 0 && !isIn(V3205,2,_01)`

B4016 [W] Occupation avant l'admission (travaux domestiques dans son propre ménage) : La donnée manque

`strlen(V3206) == 0`

B4017 [E] Occupation avant l'admission (travaux domestiques dans son propre ménage) : La donnée est invalide

`strlen(V3206) > 0 && !isIn(V3206,2,_01)`

B4018 [W] Occupation avant l'admission (en formation) : La donnée manque

`strlen(V3207) == 0`

B4019 [E] Occupation avant l'admission (en formation) : La donnée est invalide

`strlen(V3207) > 0 && !isIn(V3207,2,_01)`

B4020 [W] Occupation avant l'admission (en programme de réadaptation) : La donnée manque

`strlen(V3208) == 0`

B4021 [E] Occupation avant l'admission (en programme de réadaptation) : La donnée est invalide

`strlen(V3208) > 0 && !isIn(V3208,2,_01)`

B4022 [W] Occupation avant l'admission (AI, AVS ou autres rentes) : La donnée manque

`strlen(V3209) == 0`

B4023 [E] Occupation avant l'admission (AI, AVS ou autres rentes) : La donnée est invalide

`strlen(V3209) > 0 && !isIn(V3209,2,_01)`

B4024 [W] Occupation avant l'admission (travail dans un cadre protégé) : La donnée manque

`strlen(V3210) == 0`

B4025 [E] Occupation avant l'admission (travail dans un cadre protégé) : La donnée est invalide

`strlen(V3210) > 0 && !isIn(V3210,2,_01)`

B4026 [W] Occupation avant l'admission (inconnue) : La donnée manque

`strlen(V3211) == 0`

B4027 [E] Occupation avant l'admission (inconnue) : La donnée est invalide

`strlen(V3211) > 0 && !isIn(V3211,2,_01)`

B4040 [W] Formation scolaire ou professionnelle la plus élevée : La donnée manque

`strlen(V3212) == 0`

B4041 [E] Formation scolaire ou professionnelle la plus élevée : La donnée est invalide

`strlen(V3212) > 0 && !isIn(V3212,7,_7_AUSBILD_PSY)`

B41. Admission

B4110 [W] Séjour avant l'entrée (psychiatrie) : La donnée manque

`strlen(V3202) == 0`

ACTION : V3202="90";

B4111 [E] Séjour avant l'entrée (psychiatrie) : La donnée est invalide

`strlen(V3202) > 0 && !isIn(V3202,22,_22_AUFEIN_PSY)`

ACTION : V3202="90";

B4112 [W] Séjour avant l'entrée (psychiatrie) : Donnée plus détaillée requise (2 positions)

`strlen(V3202) > 0 && !isIn(V3202,9,_9_AUFEIN)`

B4115 [E] Séjour avant : Inconsistance entre données minimales et complément psychiatrique

`strlen(V3202) > 0 && strcmp(MB->V0202,'A') == 0 && V3202[0] != MB->V1202[0]`

B4130 [W] Instance décidant l'envoi (psychiatrie) : La donnée manque

`strlen(V3301) == 0`

ACTION : V3301="90";

B4131 [E] Instance décidant l'envoi (psychiatrie) : La donnée est invalide

`strlen(V3301) > 0 && !isIn(V3301,20,_20_EININST_PSY)`

ACTION : V3301="90";

B4132 [W] Instance décidant l'envoi (psychiatrie) : Donnée plus détaillée requise (2 positions)

`strlen(V3301) > 0 && !isIn(V3301,8,_8_EININST)`

ACTION : V3302="9";

B4135 [E] Instance décidant l'envoi : Inconsistance entre données minimales et complément psychiatrique

`strlen(V3301) > 0 && strcmp(MB->V0202,'A') == 0 && V3301[0] != MB->V1204[0]`

B4140 [W] Entrée volontaire : La donnée manque

strlen(V3302) == 0

ACTION : V3302="9";

B4141 [E] Entrée volontaire : La donnée est invalide

strlen(V3302) > 0 && !isn(V3302,3,_3_FREIWILLIGKEIT)

B4142 [W] Entrée volontaire : La donnée n'est plus nécessaire

strlen(V3302) > 0

B4150 [W] Placement à des fins d'assistance (PAFA) : La donnée manque

strlen(V3303) == 0

B4151 [E] Placement à des fins d'assistance (PAFA) : La donnée est invalide

strlen(V3303) > 0 && !isn(V3303,2,_12)

B42. Séjour

B4210 [W] Nombre de jours / nombre de consultations : La donnée manque

strlen(V3401) == 0

B4211 [E] Nombre de jours / nombre de consultations : La donnée est invalide

strlen(V3401) > 0 && (atol(V3401) < 0 || atol(V3401) > 9999)

B4220 [W] Traitement (psychiatrie) : La donnée manque

strcmp(MB->V0202,'A') == 0 && strlen(V3402) == 0

B4221 [E] Traitement (psychiatrie) : La donnée est invalide

strlen(V3402) > 0 && !isn(V3402,12,_12_BEHPY)

B4235 [H] Données supplémentaires de psychiatrie / Centre de charge : Centre de charge psychiatrie (M500) attendu

strcmp(MB->V1401,'M500') != 0

B43. Sortie

B4310 [W] Décision de sortie (psychiatrie) : La donnée manque

strlen(V3501) == 0 && strcmp(MB->V0202,'A') == 0

ACTION : V3501="90";

B4311 [E] Décision de sortie (psychiatrie) : La donnée est invalide

strlen(V3501) > 0 && !isn(V3501,9,_9_AUSENT_PSY)

ACTION : V3501="90";

B4312 [W] Décision de sortie (psychiatrie) : Donnée plus détaillée requise (2 positions)

`strlen(V3501) > 0 && !isln(V3501,7,_7_AUSENT)`

B4315 [W] Cas B/C avec MP : Il ne faut pas remplir les variables de sortie V3501 Décision de sortie, V3502 Séjour après la sortie, V3503 Prise en charge après la sortie

`strcmp(MB->V0202,'A') != 0 && (strlen(V3501) > 0 || strlen(V3502) > 0 || strlen(V3503) > 0)`

`ACTION : V3501 = "" ; V3502="" ; V3503="" ;`

B4316 [E] Décision de sortie : Inconsistance entre données minimales et complément psychiatrique

`strlen(V3501) > 0 && strcmp(MB->V0202,'A') == 0 && V3501[0] != MB->V1502[0]`

B4320 [W] Séjour après la sortie (psychiatrie) : La donnée manque

`strlen(V3502) == 0 && strcmp(MB->V0202,'A') == 0`

`ACTION : V3502="90" ;`

B4321 [E] Séjour après la sortie (psychiatrie) : La donnée est invalide

`strlen(V3502) > 0 && !isln(V3502,13,_13_AUSAUF_PSY)`

`ACTION : V3502="90" ;`

B4322 [W] Séjour après la sortie (psychiatrie) : Donnée plus détaillée requise (2 positions)

`strlen(V3502) > 0 && !isln(V3502,10,_10_AUSAUF)`

B4325 [E] Séjour après la sortie : Inconsistance entre données minimales et complément psychiatrique

`strlen(V3502) > 0 && strcmp(MB->V0202,'A') == 0 && V3502[0] != MB->V1503[0]`

B4330 [W] Prise en charge après la sortie (psychiatrie) : La donnée manque

`strlen(V3503) == 0 && strcmp(MB->V0202,'A') == 0`

`ACTION : V3503="90" ;`

B4331 [E] Prise en charge après la sortie (psychiatrie) : La donnée est invalide

`strlen(V3503) > 0 && !isln(V3503,20,_20_AUSBEH_PSY)`

`ACTION : V3503="90" ;`

B4332 [W] Prise en charge après la sortie (psychiatrie) : Donnée plus détaillée requise (2 positions)

`strlen(V3503) > 0 && !isln(V3503,8,_8_AUSBEH)`

B4335 [E] Traitement après la sortie : Inconsistance entre données minimales et complément psychiatrique

`strlen(V3503) > 0 && strcmp(MB->V0202,'A') == 0 && V3503[0] != MB->V1504[0]`

B47. Psychopharmacothérapie

B4700 [W] Psychopharmacothérapie : neuroleptiques : La donnée manque

`strlen(V3403) == 0`

B4701 [E] Psychopharmacothérapie : neuroleptiques : La donnée est invalide

`strlen(V3403) > 0 && !isln(V3403,2,_01)`

B4705 [W] Psychopharmacothérapie : neuroleptiques-dépôt : La donnée manque

`strlen(V3404) == 0`

B4706 [E] Psychopharmacothérapie : neuroleptiques-dépôt : La donnée est invalide

`strlen(V3404) > 0 && !isln(V3404,2,_01)`

B4710 [W] Psychopharmacothérapie : antidépresseurs : La donnée manque

`strlen(V3405) == 0`

B4711 [E] Psychopharmacothérapie : antidépresseurs : La donnée est invalide

`strlen(V3405) > 0 && !isln(V3405,2,_01)`

B4715 [W] Psychopharmacothérapie : tranquillisants : La donnée manque

`strlen(V3406) == 0`

B4716 [E] Psychopharmacothérapie : tranquillisants : La donnée est invalide

`strlen(V3406) > 0 && !isln(V3406,2,_01)`

B4720 [W] Psychopharmacothérapie : hypnotiques : La donnée manque

`strlen(V3407) == 0`

B4721 [E] Psychopharmacothérapie : hypnotiques : La donnée est invalide

`strlen(V3407) > 0 && !isln(V3407,2,_01)`

B4725 [W] Psychopharmacothérapie : antiépileptiques : La donnée manque

`strlen(V3408) == 0`

B4726 [E] Psychopharmacothérapie : antiépileptiques : La donnée est invalide

`strlen(V3408) > 0 && !isln(V3408,2,_01)`

B4730 [W] Psychopharmacothérapie : lithium : La donnée manque

`strlen(V3409) == 0`

B4731 [E] Psychopharmacothérapie : lithium : La donnée est invalide

strlen(V3409) > 0 && !isIn(V3409,2,_01)

B4735 [W] Psychopharmacothérapie : thérapie de substitution : La donnée manque

strlen(V3410) == 0

B4736 [E] Psychopharmacothérapie : thérapie de substitution : La donnée est invalide

strlen(V3410) > 0 && !isIn(V3410,2,_01)

B4740 [W] Psychopharmacothérapie : usage d'antagonistes : La donnée manque

strlen(V3411) == 0

B4741 [E] Psychopharmacothérapie : usage d'antagonistes : La donnée est invalide

strlen(V3411) > 0 && !isIn(V3411,2,_01)

B4745 [W] Psychopharmacothérapie : antiparkinsoniens : La donnée manque

strlen(V3412) == 0

B4746 [E] Psychopharmacothérapie : antiparkinsoniens : La donnée est invalide

strlen(V3412) > 0 && !isIn(V3412,2,_01)

B4750 [W] Psychopharmacothérapie : autres : La donnée manque

strlen(V3413) == 0

B4751 [E] Psychopharmacothérapie : autres : La donnée est invalide

strlen(V3413) > 0 && !isIn(V3413,2,_01)

B4755 [W] Psychopharmacothérapie : médic. contre les souffrances physiques : La donnée manque

strlen(V3414) == 0

B4756 [E] Psychopharmacothérapie : médic. contre les souffrances physiques : La donnée est invalide

strlen(V3414) > 0 && !isIn(V3414,2,_01)

B4900 [E] Lorsqu'un cas hospitalier A figure dans l'enregistrement M500 Psychiatrie, un enregistrement MP supplémentaire est également obligatoire

*MB->V0202 == 'A' && *MB->V1301 == '3' && strcmp(MB->V1401, 'M500') == 0 && !MB->hasMP

B5. MK

B50.

B5005 [E] Abréviation du canton dans les données supplémentaires cantonales : La donnée manque

isMKCH && strlen(V5102) == 0

B5006 [E] Abréviation du canton dans les données supplémentaires cantonales : La donnée n'est pas valable

`isMKCH && strlen(V5102) > 0 && !isIn(V5102,26,_26_KANTON)`

B51.

B5100 [E] Les données supplémentaires cantonales obligatoires manquent.

`strcmp(MB->V0202,'A') == 0 && MK->isMKGR && strcmp(V0304,'0') == 0`

B5105 [W] 2ème champ (abréviation du canton) du supplément cantonal GR : La donnée manque

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5102) == 0`

B5106 [E] 2ème champ (abréviation du canton) du supplément cantonal GR : La donnée est invalide

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5102) > 0 && !isIn(V5102,26,_26_KANTON)`

B5110 [W] 3ème champ (numéro REE) du supplément cantonal GR : La donnée manque

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5103) == 0`

B5111 [E] 3ème champ (numéro REE) du supplément cantonal GR : La donnée est invalide

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5103) > 0 && ((!isInt(V5103) || (atol(V5103) < 70000000 || atol(V5103) >= 90000000)))`

B5115 [W] 4ème champ (numéro courant) du supplément cantonal GR : La donnée manque

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5104) == 0`

B5116 [E] 4ème champ (numéro courant) du supplément cantonal GR : La donnée est invalide

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5104) > 0 && strcmp(V5104,'1.0') != 0`

B5120 [W] 5ème champ (numéro du patient) du supplément cantonal GR : La donnée manque

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5105) == 0`

B5121 [E] 5ème champ (numéro du patient) du supplément cantonal GR : La donnée est invalide

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5105) > 0 && !isAlphaNumUpperCase(V5105)`

B5125 [W] 6ème champ (DRG à facturer) du supplément cantonal GR : La donnée manque

`strcmp(MB->V0202,'A') == 0 && !isIn(MB->V1401,3,_3_HKST_PSY_REHA) && isMKGR && strlen(V5106) == 0`

B5130 [W] 7ème champ (version grouper DRG) du supplément cantonal GR : La donnée manque

`strcmp(MB->V0202,'A') == 0 && isMKGR && !isIn(MB->V1401,3,_3_HKST_PSY_REHA) && strlen(V5107) == 0`

B5131 [E] 7ème champ (version grouper DRG) du supplément cantonal GR : La donnée est invalide

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5107) > 0 && !isIn(V5107,8,_8_SDRG_VERSION)`

B5135 [W] 8ème champ (CW facturé) du supplément cantonal GR : La donnée manque

```
strcmp(MB->V0202,'A') == 0 && isMKGR && !isIn(MB->V1401,3,_3_HKST_PSY_REHA) && strlen(V5108) == 0
```

B5136 [W] 8ème champ (CW facturé) du supplément cantonal GR : La donnée manque

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5108) > 0 && (!isFloat(V5108) || atof(V5108) < 0 || atof(V5108) > 9999)
```

B5140 [W] 9ème champ (durée du séjour) du supplément cantonal GR : La donnée manque

```
strcmp(MB->V0202,'A') == 0 && isMKGR && !isIn(MB->V1401,3,_3_HKST_PSY_REHA) && strlen(V5109) == 0
```

B5141 [E] 9ème champ (durée du séjour) du supplément cantonal GR : La donnée est invalide

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5109) > 0 && (!isInt(V5109) || atol(V5109) < 0 || atol(V5109) > 9999)
```

B5145 [W] 10ème champ (numéro OFSP de la caisse maladie) du supplément cantonal GR : La donnée manque

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5110) == 0
```

B5146 [W] 10ème champ (numéro OFSP de la caisse maladie) du supplément cantonal GR : La donnée est invalide

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5110) > 0 && (strcmp(V5110,'9999') == 0 || (!V5110_vsnr.inIndex && !isIn(V5110,5,_5_VSNR)))
```

B5150 [W] 11ème champ (baserate) du supplément cantonal GR : La donnée manque

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5111) == 0
```

B5151 [E] 11ème champ (baserate) du supplément cantonal GR : La donnée est invalide

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5111) > 0 && (!isInt(V5111) || atol(V5111) < 0 || atol(V5111) > 99999)
```

B5155 [W] 12ème champ (rémunérations sup.) du supplément cantonal GR : La donnée manque

```
strcmp(MB->V0202,'A') == 0 && isMKGR && !isIn(MB->V1401,3,_3_HKST_PSY_REHA) && strlen(V5112) == 0
```

B5156 [E] 12ème champ (rémunérations sup.) du supplément cantonal GR : La donnée est invalide

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5112) > 0 && (!isFloat(V5112) || atof(V5112) < 0 || atof(V5112) > 999999)
```

B5160 [W] 13ème champ (numéro du cas) du supplément cantonal GR : La donnée manque

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5113) == 0
```

B5161 [E] 13ème champ (numéro du cas) du supplément cantonal GR : le champ est limité à 50 caractères

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5113) > 50
```

B5162 [E] 13ème champ (numéro du cas) du supplément cantonal GR : caractères spéciaux non valides

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5113) > 0 && !isAlphaNumUpperCase(V5113)
```

B5182 [W] 14ème champ (champ de réserve 2) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5114) > 0
```

B5183 [W] 15ème champ (champ de réserve 3) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5115) > 0
```

B5184 [W] 16ème champ (champ de réserve 4) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5116) > 0
```

B5185 [W] 17ème champ (champ de réserve 5) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5117) > 0
```

B5186 [W] 18ème champ (champ de réserve 6) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5118) > 0
```

B5187 [W] 19ème champ (champ de réserve 7) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5119) > 0
```

B5188 [W] 20ème champ (champ de réserve 8) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5120) > 0
```

B5189 [W] 21ème champ (champ de réserve 9) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5121) > 0
```

B5190 [W] 22ème champ (champ de réserve 10) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5122) > 0
```

B5191 [W] 23ème champ (champ de réserve 11) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

```
strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5123) > 0
```

B5192 [W] 24ème champ (champ de réserve 12) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5124) > 0`

B5193 [W] 25ème champ (champ de réserve 13) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5125) > 0`

B5194 [W] 26ème champ (champ de réserve 14) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5126) > 0`

B5195 [W] 27ème champ (champ de réserve 15) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5127) > 0`

B5196 [W] 28ème champ (champ de réserve 16) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5128) > 0`

B5197 [W] 29ème champ (champ de réserve 17) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5129) > 0`

B5198 [W] 30ème champ (champ de réserve 18) du supplément cantonal GR : contient des données, malgré le fait que le champ n'est pas encore défini

`strcmp(MB->V0202,'A') == 0 && isMKGR && strlen(V5130) > 0`

B52.

B5205 [E] 2ème champ (canton) : La donnée manque

`isMKLU && strlen(V5102) == 0`

B5206 [E] Canton : La donnée n'est pas valable

`isMKLU && strlen(V5102) > 0 && !isIn(V5102,6,_6_KTLU)`

B5207 [E] Canton de l'enregistrement MB : La donnée n'est pas valable

`isMKLU && strlen(MB->V0104) > 0 && !isIn(MB->V0104,6,_6_KTLU)`

B5208 [E] Differences des cantons dans les enregistrements MB et MK : La donnée n'est pas valable

`isMKLU && strlen(MB->V0104) > 0 && strlen(V5102) > 0 && strcmp(MB->V0104,V5102) != 0`

B5210 [E] 3ème champ (numéro REE) : La donnée manque

`isMKLU && strlen(V5103) == 0`

B5215 [E] 4ème champ (numéro du patient) : La donnée manque

isMKLU && strlen(V5104) == 0

B53.

B5302 [X]

strlen(MK->V6202) == 0

B5303 [X]

strlen(MK->V6203) == 0

B5304 [X]

strlen(MK->V6204) == 0

B5305 [X]

strlen(MK->V6205) == 0

B5306 [X]

strlen(MK->V6206) == 0

B5307 [X]

strlen(MK->V6207) == 0

B5350 [X]

strlen(MK->VLM03) == 0

B6. Identification des réhospitalisations

B6640 [E] Réadmissions ultérieures : La donnée est invalide

strlen(V4741) > 0 && !(strcmp(V4741,'0') == 0 || strcmp(V4741,'1') == 0)

B6900 [E] 1ère interruption, sortie : La donnée concernant la date n'est pas valable

!dateV4701.isNull && !dateV4701.isYYYYMMDD

B6901 [E] 1ère réadmission : La donnée concernant la date n'est pas valable

!dateV4702.isNull && !dateV4702.isYYYYMMDD

B6902 [E] 2ème interruption, sortie : La donnée concernant la date n'est pas valable

!dateV4711.isNull && !dateV4711.isYYYYMMDD

B6903 [E] 2ème réadmission : La donnée concernant la date n'est pas valable

!dateV4712.isNull && !dateV4712.isYYYYMMDD

B6904 [E] 3ème interruption, sortie : la donnée concernant la date n'est pas valable

!dateV4721.isNull && !dateV4721.isYYYYMMDD

B6905 [E] 3ème réadmission : la donnée concernant la date n'est pas valable

!dateV4722.isNull && !dateV4722.isYYYYMMDD

B6906 [E] 4ème interruption, sortie : La donnée concernant la date n'est pas valable

!dateV4731.isNull && !dateV4731.isYYYYMMDD

B6907 [E] 4ème réadmission : La donnée concernant la date n'est pas valable

!dateV4732.isNull && !dateV4732.isYYYYMMDD

B6910 [E] 1ère interruption, sortie (4.7.V01) : Elle a lieu avant la date d'admission

strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4701.isNull && MB->dateV1201.DateValue > dateV4701.DateValue

B6911 [E] La 1ère réadmission (4.7.V02) a lieu avant la date d'admission

strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4702.isNull && MB->dateV1201.DateValue > dateV4702.DateValue

B6912 [E] 2ème interruption, sortie (4.7.V11) : Elle a lieu avant la date d'admission

strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4711.isNull && MB->dateV1201.DateValue > dateV4711.DateValue

B6913 [E] La 2ème réadmission (4.7.V12) a lieu avant la date d'admission

strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4712.isNull && MB->dateV1201.DateValue > dateV4712.DateValue

B6914 [E] 3ème interruption, sortie (4.7.V21) : Elle a lieu avant la date d'admission

strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4721.isNull && MB->dateV1201.DateValue > dateV4721.DateValue

B6915 [E] La 3ème réadmission (4.7.V22) a lieu avant la date d'admission

strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4722.isNull && MB->dateV1201.DateValue > dateV4722.DateValue

B6916 [E] 4ème interruption, sortie (4.7.V31) : Elle a lieu avant la date d'admission

strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4731.isNull && MB->dateV1201.DateValue > dateV4731.DateValue

B6917 [E] La 4ème réadmission (4.7.V32) a lieu avant la date d'admission

strcmp(MB->V0202,'A') == 0 && !MB->dateV1201.isNull && !dateV4732.isNull && MB->dateV1201.DateValue > dateV4732.DateValue

B6920 [E] 1ère interruption, sortie (4.7.V01) : Elle a lieu après la date de sortie

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4701.isNull && dateV4701.DateValue > MB->dateV1501.DateValue`

B6921 [E] La 1ère réadmission (4.7.V02) a lieu après la date de sortie

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4702.isNull && dateV4702.DateValue > MB->dateV1501.DateValue`

B6922 [E] 2ème interruption, sortie (4.7.V11) : Elle a lieu après la date de sortie

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4711.isNull && dateV4711.DateValue > MB->dateV1501.DateValue`

B6923 [E] La 2ème réadmission (4.7.V12) a lieu après la date de sortie

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4712.isNull && dateV4712.DateValue > MB->dateV1501.DateValue`

B6924 [E] 3ème interruption, sortie (4.7.V21) : Elle a lieu après la date de sortie

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4721.isNull && dateV4721.DateValue > MB->dateV1501.DateValue`

B6925 [E] La 3ème réadmission (4.7.V22) a lieu après la date de sortie

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4722.isNull && dateV4722.DateValue > MB->dateV1501.DateValue`

B6926 [E] 4ème interruption, sortie (4.7.V31) : Elle a lieu après la date de sortie

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4731.isNull && dateV4731.DateValue > MB->dateV1501.DateValue`

B6927 [E] La 4ème réadmission (4.7.V32) a lieu après la date de sortie

`strcmp(MB->V0202,'A') == 0 && !MB->dateV1501.isNull && !dateV4732.isNull && dateV4732.DateValue > MB->dateV1501.DateValue`

B6930 [E] Première interruption, sortie : Elle doit avoir lieu avant la seconde interruption

`strcmp(MB->V0202,'A') == 0 && !dateV4701.isNull && !dateV4711.isNull && dateV4701.DateValue > dateV4711.DateValue`

B6931 [E] Seconde interruption, sortie : Elle doit avoir lieu avant la troisième interruption

`strcmp(MB->V0202,'A') == 0 && !dateV4711.isNull && !dateV4721.isNull && dateV4711.DateValue > dateV4721.DateValue`

B6932 [E] Troisième interruption, sortie : Elle doit avoir lieu avant la quatrième interruption

`strcmp(MB->V0202,'A') == 0 && !dateV4721.isNull && !dateV4731.isNull && dateV4721.DateValue > dateV4731.DateValue`

B6940 [E] La première réadmission doit avoir lieu avant la seconde réadmission

`strcmp(MB->V0202,'A') == 0 && !dateV4702.isNull && !dateV4712.isNull && dateV4702.DateValue > dateV4712.DateValue`

B6941 [E] La seconde réadmission doit avoir lieu avant la troisième réadmission

`strcmp(MB->V0202,'A') == 0 && !dateV4712.isNull && !dateV4722.isNull && dateV4712.DateValue > dateV4722.DateValue`

B6942 [E] La troisième réadmission doit avoir lieu avant la quatrième réadmission

`strcmp(MB->V0202,'A') == 0 && !dateV4722.isNull && !dateV4732.isNull && dateV4722.DateValue > dateV4732.DateValue`

B6950 [E] Raison de la 1ère réadmission : La donnée n'est pas valable

`strlen(V4703) > 0 && !isIn(V4703,4,_1239)`

B6951 [E] Raison de la 2ème réadmission : La donnée n'est pas valable

`strlen(V4713) > 0 && !isIn(V4713,4,_1239)`

B6952 [E] Raison de la 3ème réadmission : La donnée n'est pas valable

`strlen(V4723) > 0 && !isIn(V4723,4,_1239)`

B6953 [E] Raison de la 4ème réadmission : La donnée n'est pas valable

`strlen(V4733) > 0 && !isIn(V4733,4,_1239)`

B6960 [E] Lorsqu'une 1ère sortie est mentionnée, la 1ère réadmission doit l'être pareillement

`strlen(V4701) > 0 && strlen(V4702) == 0`

B6961 [E] Lorsqu'une 2ème sortie est mentionnée, la 2ème réadmission doit l'être pareillement

`strlen(V4711) > 0 && strlen(V4712) == 0`

B6962 [E] Lorsqu'une 3ème sortie est mentionnée, la 3ème réadmission doit l'être pareillement

`strlen(V4721) > 0 && strlen(V4722) == 0`

B6963 [E] Lorsqu'une 4ème sortie est mentionnée, la 4ème réadmission doit l'être pareillement

`strlen(V4731) > 0 && strlen(V4732) == 0`

B6970 [E] Lorsqu'une 1ère réadmission est mentionnée, la raison de celle-ci doit l'être pareillement

`strlen(V4702) > 0 && strlen(V4703) == 0`

B6971 [E] Lorsqu'une 2ème réadmission est mentionnée, la raison de celle-ci doit l'être pareillement

`strlen(V4712) > 0 && strlen(V4713) == 0`

B6972 [E] Lorsqu'une 3ème réadmission est mentionnée, la raison de celle-ci doit l'être pareillement

`strlen(V4722) > 0 && strlen(V4723) == 0`

B6973 [E] Lorsqu'une 4ème réadmission est mentionnée, la raison de celle-ci doit l'être pareillement

`strlen(V4732) > 0 && strlen(V4733) == 0`

B6975 [E] Le cas présente une 1ère interruption, sortie, ayant eu lieu au cours de l'année précédente, ainsi qu'une sortie dans l'année courante. Cet enregistrement d'une réhospitalisation n'est pas conforme aux règles de SwissDRG.

`strcmp(MB->V0202,'A') == 0 && dateV4701.isYYYYMMDD && dateV4701.Year != MB->dateV1501.Year`

B6976 [E] La durée des vacances administratives correspond à la longueur de l'absence entre sorties et réadmissions. L'absence a-t-elle été saisie deux fois ?

`dateV4701.hasDate && UrlaubT > 0 && UrlaubT == ABS`

B6977 [E] La durée du séjour calculée se monte à moins d'un jour.

`strcmp(MB->V0202,'A') == 0 && LOS < 1`

B6978 [E] Une réadmission remonte à plus de dix-huit jours avant la première sortie. Veuillez contrôler ce cas de réhospitalisation.

`(dateV4702.hasDate && dateV4702.DateValue - dateV4701.DateValue > 18) || (dateV4712.hasDate && dateV4712.DateValue - dateV4701.DateValue > 18) || (dateV4722.hasDate && dateV4722.DateValue - dateV4701.DateValue > 18) || (dateV4732.hasDate && dateV4732.DateValue - dateV4701.DateValue > 18)`

B6980 [E] La 1ère réadmission a eu lieu plus de 18 jours après la 1ère interruption (sortie).

`dateV4701.isYYYYMMDD && dateV4702.isYYYYMMDD && dateV4702.DateValue - dateV4701.DateValue > 18`

B6981 [E] La 2ème réadmission a eu lieu plus de 18 jours après la 1ère interruption (sortie).

`dateV4701.isYYYYMMDD && dateV4712.isYYYYMMDD && dateV4712.DateValue - dateV4701.DateValue > 18`

B6982 [E] La 3ème réadmission a eu lieu plus de 18 jours après la 1ère interruption (sortie).

`dateV4701.isYYYYMMDD && dateV4722.isYYYYMMDD && dateV4722.DateValue - dateV4701.DateValue > 18`

B6983 [E] La 4ème réadmission a eu lieu plus de 18 jours après la 1ère interruption (sortie).

`dateV4701.isYYYYMMDD && dateV4732.isYYYYMMDD && dateV4732.DateValue - dateV4701.DateValue > 18`

B6984 [E] 1ère interruption, sortie/1ère réadmission : La donnée n'est pas plausible.

`strcmp(MB->V0202,'A') == 0 && (strlen(V4701) > 0 || strlen(V4702) > 0) && (!dateV4701.isYYYYMMDD || !dateV4702.isYYYYMMDD || dateV4701.DateValue < MB->dateV1201.DateValue || dateV4702.DateValue < dateV4701.DateValue || dateV4702.DateValue > MB->dateV1501.DateValue)`

B6985 [E] 2ème interruption, sortie/2ème réadmission : La donnée n'est pas plausible.

`strcmp(MB->V0202,'A') == 0 && (strlen(V4711) > 0 || strlen(V4712) > 0) && (!dateV4711.isYYYYMMDD || !dateV4712.isYYYYMMDD || dateV4711.DateValue < dateV4702.DateValue || dateV4712.DateValue < dateV4711.DateValue || dateV4712.DateValue > MB->dateV1501.DateValue)`

B6986 [E] 3ème interruption, sortie/3ème réadmission : La donnée n'est pas plausible.

```
strcmp(MB->V0202,'A') == 0 && (strlen(V4721) > 0 || strlen(V4722) > 0) && ( !dateV4721.isYYYYMMDD || !dateV4722.isYYYYMMDD  
|| dateV4721.DateValue < dateV4712.DateValue || dateV4722.DateValue < dateV4721.DateValue || dateV4722.DateValue >  
MB->dateV1501.DateValue)
```

B6987 [E] 4ème interruption, sortie/4ème réadmission : La donnée n'est pas plausible.

```
strcmp(MB->V0202,'A') == 0 && (strlen(V4731) > 0 || strlen(V4732) > 0) && ( !dateV4731.isYYYYMMDD || !dateV4732.isYYYYMMDD  
|| dateV4731.DateValue < dateV4722.DateValue || dateV4732.DateValue < dateV4731.DateValue || dateV4732.DateValue >  
MB->dateV1501.DateValue)
```

C. Erreurs formelles

C0. Problèmes structurels

C0000 [E] Nombre d'enregistrements livrés : Le nombre indiqué dans l'annonce de livraison a été dépassé

```
LineNumberInMX > atol(MX->V0008)
```

C0220 [E] MB : Test cas statistique A : La sortie doit se situer dans la période de relevé

```
strcmp(V0202,'A') == 0 && dateV1501.hasDate && (dateV1501.DateValue < MX->dateV0006.DateValue || dateV1501.DateValue > MX->dateV0007.DateValue)
```

```
ACTION : doWriteRec==FALSE;
```

C0301 [E] Données des nouveau-nés : Enregistrement supplémentaire trouvé sans annonce dans les données minimales

```
strcmp(MB->V0301,'0') == 0
```

```
ACTION : MB->V0301="1";
```

C0320 [E] MN : Test cas statistique A : La sortie doit se situer dans la période de relevé

```
strcmp(MB->V0202,'A') == 0 && MB->dateV1501.hasDate && (MB->dateV1501.DateValue < MX->dateV0006.DateValue || MB->dateV1501.DateValue > MX->dateV0007.DateValue)
```

C0400 [E] Données supplémentaires de psychiatrie : L'enregistrement supplémentaire manque

```
strcmp(V0302,'1') == 0 && !hasMP
```

```
ACTION : V0302="0";
```

C0401 [E] Données supplémentaires de psychiatrie : Enregistrement supplémentaire trouvé sans annonce dans les données minimales

```
strcmp(MB->V0302,'0') == 0
```

C0420 [E] MP : Test cas statistique A : La sortie doit se situer dans la période de relevé

```
strcmp(MB->V0202,'A') == 0 && MB->dateV1501.hasDate && (MB->dateV1501.DateValue < MX->dateV0006.DateValue || MB->dateV1501.DateValue > MX->dateV0007.DateValue)
```

```
ACTION : MB->V0302="0"; MB->hasMP==FALSE;
```

C0600 [E] Données supplémentaires cantonales : L'enregistrement supplémentaire manque

```
strcmp(V0304,'1') == 0 && !hasMK
```

```
ACTION : V0304="0";
```


C0601 [E] Données supplémentaires cantonales : Enregistrement supplémentaire trouvé sans annonce dans les données minimales

```
strcmp(MB->V0304,'0') == 0
```

```
ACTION : MB->V0304="1" ;
```

C0620 [E] MK : Test cas statistique A : La sortie doit se situer dans la période de relevé

```
strcmp(MB->V0202,'A') == 0 && MB->dateV1501.hasDate && (MB->dateV1501.DateValue < MX->dateV0006.DateValue  
|| MB->dateV1501.DateValue > MX->dateV0007.DateValue)
```

C0700 [E] Données supplémentaires par groupes de patients : L'enregistrement supplémentaire manque

```
strcmp(V0303,'1') == 0 && !hasMD
```

```
ACTION : V0303="0" ;
```

C0701 [E] Données supplémentaires par groupes de patients : Enregistrement supplémentaire trouvé sans annonce dans les données minimales

```
strcmp(MB->V0303,'0') == 0
```

```
ACTION : MB->V0303="1" ;
```

C0710 [E] Données supplémentaires par groupes de patients : L'enregistrement supplémentaire MD manque. Il est obligatoire.

```
strcmp(V0202,'A') == 0 && strcmp(V1301,'3') == 0 && !hasMD
```

C0720 [E] MD : Test cas statistique A : La sortie doit se situer dans la période de relevé

```
strcmp(MB->V0202,'A') == 0 && MB->dateV1501.hasDate && (MB->dateV1501.DateValue < MX->dateV0006.DateValue  
|| MB->dateV1501.DateValue > MX->dateV0007.DateValue)
```

C0900 [E] Caractère spécial non autorisé dans le MX

```
checkChars() != -1
```

C0901 [E] Caractère spécial non autorisé dans le MB

```
checkChars() != -1
```

C0902 [E] Caractère spécial non autorisé dans le MN

```
checkChars() != -1
```

C0903 [E] Caractère spécial non autorisé dans le MP

```
checkChars() != -1
```

C0904 [E] Caractère spécial non autorisé dans le MD

```
checkChars() != -1
```

C0905 [E] Caractère spécial non autorisé dans le MK

checkChars() != -1

C1. Problèmes MX

C10. MX Missing

C1010 [E] Numéro REE : La donnée manque

strlen(V0002) == 0

C1020 [E] Abréviation de la clinique : La donnée manque

strlen(V0003) == 0

ACTION : V0003="Dummy";

C1021 [E] Abréviation de la clinique : La donnée ne peut avoir que huit caractères au maximum

strlen(V0003) > 8

ACTION : V0003[8]=0;

C1030 [E] Date de l'exportation : La donnée manque

dateV0004.isNull

C1031 [E] Date de l'exportation : La donnée concernant la date est invalide

!(dateV0004.isNull || dateV0004.isYYYYMMDD)

ACTION : if (strlen(V0004) > 8) V0004[8]=0;

C1040 [E] Numéro de la livraison : La donnée manque

strlen(V0005) == 0

ACTION : V0005="1";

C1041 [E] Numéro de la livraison : La donnée est invalide

strlen(V0005) > 0 && ((!isInt(V0005) || atoi(V0005) < 1 || atoi(V0005) > 99999))

ACTION : V0005="1";

C1050 [E] Période de l'annonce du : La donnée manque

dateV0006.isNull

C1051 [E] Période de l'annonce du : La donnée concernant la date est invalide

!(dateV0006.isNull || dateV0006.isYYYYMMDD)

ACTION : if (strlen(V0006) > 8) V0006[8]=0;

C1060 [E] Période de l'annonce jusqu'à : La donnée manque

dateV0007.isNull

C1061 [E] Période de l'annonce jusqu'à : La donnée concernant la date est invalide

!(dateV0007.isNull || dateV0007.isYYYYMMDD)

ACTION : if (strlen(V0007) > 8) V0007[8]=0;

C1070 [E] Nombre d'enregistrements : La donnée manque

strlen(V0008) == 0

C1071 [E] Nombre d'enregistrements : La donnée est invalide

strlen(V0008) != 0 && ((!isInt(V0008) || atol(V0008) < 3 || atol(V0008) > 99999999))

C1080 [E] Champ clé C : La donnée manque

strlen(V0009) == 0

C1081 [E] Champ clé C : Le type de donnée est erroné (non hexadécimal)

strlen(V0009) != 0 && !isHex(V0009)

C1082 [E] Champ clé C : La longueur n'est pas correcte

strlen(V0009) > 0 && strlen(V0009) != 256

C1091 [E] Annonce de livraison : Contradiction entre la période d'annonce de, jusqu'à et y compris la date de l'exportation

dateV0007.DateValue < dateV0006.DateValue || dateV0004.DateValue < dateV0007.DateValue

C1210 [E] Pseudoenregistrement de la personne fictive 'Meier Hans' : N'est autorisé que le remplissage des deux champs Type d'enregistrement et Code de liaison anonyme

strlen(V0102) > 0 || strlen(V0103) > 0 || strlen(V0104) > 0 || strlen(V0202) > 0 || strlen(V0301) > 0 || strlen(V0302) > 0 || strlen(V0303) > 0 || strlen(V0304) > 0 || strlen(V1101) > 0 || strlen(V1102) > 0 || strlen(V1103) > 0 || strlen(V1104) > 0 || strlen(V1105) > 0 || strlen(V1201) > 0 || strlen(V1202) > 0 || strlen(V1203) > 0 || strlen(V1204) > 0 || strlen(V1301) > 0 || strlen(V1302) > 0 || strlen(V1303) > 0 || strlen(V1304) > 0 || strlen(V1401) > 0 || strlen(V1402) > 0 || strlen(V1501) > 0 || strlen(V1502) > 0 || strlen(V1503) > 0 || strlen(V1504) > 0 || strlen(DIA[0].Key) > 0

C1240 [E] Pseudoenregistrement de la personne fictive 'Meier Hans' : Le code de liaison manque dans l'enregistrement test

strlen(V0201) == 0

C1241 [E] Pseudoenregistrement de la personne fictive 'Meier Hans' : Le code de liaison n'est pas hexadécimal dans l'enregistrement test

strlen(V0201) > 0 && !isHex(V0201)

C1242 [E] Code de liaison anonyme : La donnée est invalide car reposant sur des valeurs manquantes ou illégales au niveau du nom, prénom, date de naissance (complète) ou sexe, avant l'anonymisation.

strcmp(V0201,'0000000000000000') == 0

C1243 [E] Pseudoenregistrement de la personne fictive 'Meier Hans' : Le nombre de caractères du code de liaison est incorrect

`strlen(V0201) > 0 && strlen(V0201) != 16`

C2. MB

C2010 [E] Numéro REE : La donnée manque

`strlen(V0102) == 0`

C2015 [E] Numéro REE : Inconsistance avec l'indication dans l'annonce de livraison

`strlen(V0102) > 0 && strcmp(V0102,MX->V0002) != 0`

C2020 [E] Site : La donnée manque

`strlen(V0103) == 0`

C2021 [E] Site : La donnée n'est pas valable

`strlen(V0103) > 0 && !isn (V0103,36, _NOGA)`

`ACTION : V0103="" ;`

C2022 [E] Le champ Site est limité à 2 caractères

`strlen(V0103) > 2`

C2030 [E] Canton : La donnée manque

`strlen(V0104) == 0`

C2031 [E] Canton : La donnée est invalide

`strlen(V0104) > 0 && !isn (V0104,26, _26_ KANTON)`

C2040 [E] Code de liaison anonyme : La donnée manque

`strlen(V0201) == 0`

C2041 [E] Code de liaison anonyme : Le nombre de caractères est incorrect

`strlen(V0201) > 0 && strlen(V0201) != 16`

C2042 [E] Code de liaison anonyme : La donnée n'est pas hexadécimale

`strlen(V0201) > 0 && !isHex(V0201)`

C2045 [E] Code de liaison anonyme : La donnée est invalide car reposant sur des valeurs manquantes ou illégales au niveau du nom, prénom, date de naissance (complète) ou sexe, avant l'anonymisation.

`strcmp(V0201,'0000000000000000') == 0`

C2050 [E] Marque distinctive du cas statistique : La donnée manque

strlen(V0202) == 0

C2051 [E] Marque distinctive du cas statistique : La donnée est invalide

strlen(V0202) > 0 && !isIn(V0202,3, _ABC)

C2060 [E] Données des nouveau-nés : La donnée manque

strlen(V0301) == 0

ACTION : V0301="0" ;

C2061 [E] Données des nouveau-nés : La donnée est invalide

strlen(V0301) > 0 && !isIn(V0301,2, _01)

ACTION : V0301="0" ;

C2070 [E] Données supplémentaires de psychiatrie : La donnée manque

strlen(V0302) == 0

ACTION : V0302="0" ;

C2071 [E] Données supplémentaires de psychiatrie : La donnée est invalide

strlen(V0302) > 0 && !isIn(V0302,2, _01)

ACTION : V0302="0" ;

C2080 [E] Données supplémentaires par groupes de patients : La donnée manque

strlen(V0303) == 0

ACTION : V0303="0" ;

C2081 [E] Données supplémentaires par groupes de patients : La donnée est invalide

strlen(V0303) > 0 && !isIn(V0303,2, _01)

ACTION : V0303="0" ;

C2090 [E] Données supplémentaires cantonales : La donnée manque

strlen(V0304) == 0

ACTION : V0304="0" ;

C2091 [E] Données supplémentaires cantonales : La donnée est invalide

strlen(V0304) > 0 && !isIn(V0304,2, _01)

ACTION : V0304="0" ;

C2111 [H] Date de naissance : La longueur est de huit chiffres

!dateV1102.isNull && strlen(V1102) != 8

ACTION : if (strlen(V1102) > 8) V1102[8]=0;

C2210 [W] Le diagnostic dans le MB doit correspondre au diagnostic dans le MD

strcmp(MB->V0202,'A') == 0 && strcmp(MB->DIA[i].Key,DIA[i].Key) != 0

C2220 [W] Le traitement dans le MB doit correspondre au traitement dans le MD

strcmp(MB->V0202,'A') == 0 && strncmp(BEH[i].Key,MB->BEH[i].Key,5) != 0 && (!MB->BEH[i].inIndex || strcmp(MB->BEH[i].Key,BEH[i].Key,strlen(MB->BEH[i].Key)))

C2222 [E] Le début du traitement principal dans le MD doit correspondre au début du traitement principal dans le MB

strcmp(MB->V0202,'A') == 0 && strcmp(BEH[0].dateDatum.Value,MB->V1702) != 0

C2320 [E] Test cas statistique B : L'admission doit se situer dans la période de relevé

V0202[0] == 'B' && dateV1201.hasDate && (dateV1201.DateValue < MX->dateV0006.DateValue || dateV1201.DateValue > MX->dateV0007.DateValue)

C2321 [E] Test cas statistique B : La date de sortie ne peut se situer avant la fin de la période de relevé

strcmp(V0202,'B') == 0 && dateV1501.hasDate && dateV1501.DateValue < MX->dateV0007.DateValue

C2322 [H] Test cas statistique B : Les données ne sont demandées que jusqu'à la variable 1.4.V02 (prise en charge des soins de base)

V0202[0] == 'B' && (strlen(V1501) > 0 || strlen(V1502) > 0 || strlen(V1503) > 0 || strlen(V1504) > 0 || DIACount > 0 || BEHCount > 0)

ACTION : V1501=""; V1502=""; V1503=""; V1504=""; V1702=""; for (int di=0 ; di<=DIACount ;di++) DIA[di].Key=""; for (int bi=0 ; bi<=BEHCount ;bi++) BEH[bi].Key="";

C2330 [E] Test cas statistique C : L'admission doit se situer avant la période de relevé

strcmp(V0202,'C') == 0 && dateV1201.hasDate && dateV1201.DateValue > MX->dateV0006.DateValue

C2331 [E] Test cas statistique C : La date de sortie ne peut se situer avant la fin de la période de relevé

V0202[0] == 'C' && dateV1501.hasDate && dateV1501.DateValue < MX->dateV0007.DateValue

C2332 [H] Test cas statistique C : Les données de sortie ne sont pas demandées

strcmp(V0202,'C') == 0 && (strlen(V1501) > 0 || strlen(V1502) > 0 || strlen(V1503) > 0 || strlen(V1504) > 0)

ACTION : V1501=""; V1502=""; V1503=""; V1504=""; V1702="";

C7. MD

C7010 [E] Localisation de l'hôpital : La donnée manque

strlen(V4102) == 0

C7011 [W] Localisation de l'hôpital : La donnée n'est pas valable (numéro postal requis).

`!V4102_plz.isNull && !V4102_plz.inIndex`

C7029 [E] Le champ de réserve concernant la localisation de l'hôpital n'est pas encore défini. Il doit être vide.

`strlen(V4103) > 0`

C7051 [E] Numéro du cas (statistique des coûts par cas) : le champ est limité à 16 caractères

`strlen(V4601) > 16`

C7052 [E] Numéro du cas (statistique des coûts par cas) : caractères spéciaux non valides

`strlen(V4601) > 0 && !isAlphaNumUpperCase(V4601)`

C7061 [W] Le champ Statut DRG est limité à 1 caractère

`strlen(V4801) > 1`

C7100 [E] Les codes des diagnostics doivent être listés les uns après les autres. Les champs CIM vides ('lacunes') ne sont pas autorisés dans les codes transmis.

`i > 0 && !DIA[i].isNull && ((i < 3 && DIA[0].isNull) || (i > 2 && DIA[i-1].isNull))`

C7200 [E] Les codes des traitements doivent être listés les uns après les autres. Les champs CHOP vides ('lacunes') ne sont pas autorisés dans les codes transmis.

`i > 0 && !BEH[i].isNull && BEH[i-1].isNull`

C7301 [W] Le champ Médicament 1 est limité à 50 caractères

`strlen(V4802) > 50`

C7302 [W] Le champ Médicament 2 est limité à 50 caractères

`strlen(V4803) > 50`

C7303 [W] Le champ Médicament 3 est limité à 50 caractères

`strlen(V4804) > 50`

C7304 [W] Le champ Médicament 4 est limité à 50 caractères

`strlen(V4805) > 50`

C7305 [W] Le champ Médicament 5 est limité à 50 caractères

`strlen(V4806) > 50`

C7306 [W] Le champ Médicament 6 est limité à 50 caractères

`strlen(V4807) > 50`

C7307 [W] Le champ Médicament 7 est limité à 50 caractères

`strlen(V4808) > 50`

C7308 [W] Le champ Médicament 8 est limité à 50 caractères

`strlen(V4809) > 50`

C7309 [W] Le champ Médicament 9 est limité à 50 caractères

`strlen(V4810) > 50`

C7310 [W] Le champ Médicament 10 est limité à 50 caractères

`strlen(V4811) > 50`

C7311 [W] Le champ Médicament 11 est limité à 50 caractères

`strlen(V4812) > 50`

C7312 [W] Le champ Médicament 12 est limité à 50 caractères

`strlen(V4813) > 50`

C7313 [W] Le champ Médicament 13 est limité à 50 caractères

`strlen(V4814) > 50`

C7314 [W] Le champ Médicament 14 est limité à 50 caractères

`strlen(V4815) > 50`

C7901 [E] Le champ de réserve 1 SSMII n'est pas encore défini. Il doit être vide.

`strlen(V4405) > 0`

C7902 [E] Le champ de réserve 2 SSMII n'est pas encore défini. Il doit être vide.

`strlen(V4406) > 0`

C7910 [E] Le 16ème champ de réserve n'est pas encore défini. Il doit être vide.

`strlen(V4816) > 0`

C7911 [E] Le 16ème champ de réserve est limité à 50 caractères

`strlen(V4816) > 50`

C7920 [E] Le 17ème champ de réserve n'est pas encore défini. Il doit être vide.

`strlen(V4817) > 0`

C7921 [E] Le 17ème champ de réserve est limité à 50 caractères

`strlen(V4817) > 50`

C7930 [E] Le 18ème champ de réserve n'est pas encore défini. Il doit être vide.

`strlen(V4818) > 0`

C7931 [E] Le 18ème champ de réserve est limité à 50 caractères

`strlen(V4818) > 50`

C7940 [E] Le 19ème champ de réserve n'est pas encore défini. Il doit être vide.

`strlen(V4819) > 0`

C7941 [E] Le 19ème champ de réserve est limité à 50 caractères

`strlen(V4819) > 50`

C7950 [E] Le 20ème champ de réserve n'est pas encore défini. Il doit être vide.

`strlen(V4820) > 0`

C7951 [E] Le 20ème champ de réserve est limité à 50 caractères

`strlen(V4820) > 50`

D. Indicateurs

D0. Messages

D0000 [C] TextF

$1 == 1$

D0001 [C] Enregistrement contient des erreurs

$c_ECount > 0$

D0002 [C] Enregistrement contient des avertissements

$c_WCount > 0$

D0003 [C] Enregistrement contient des indications

$c_HCount > 0$

D0004 [C] Enregistrement contient des erreurs ou des avertissements

$c_ECount + c_WCount > 0$

D0005 [C] Enregistrement contient des erreurs, des avertissements ou des indications

$c_ECount + c_WCount + c_HCount > 0$

D0101 [C] Enregistrement contient des erreurs

$c_ECount > 0$

D0102 [C] Enregistrement contient des avertissements

$c_WCount > 0$

D0103 [C] Enregistrement contient des indications

$c_HCount > 0$

D0104 [C] Enregistrement contient des erreurs ou des avertissements

$c_ECount + c_WCount > 0$

D0105 [C] Enregistrement contient des erreurs, des avertissements ou des indications

$c_ECount + c_WCount + c_HCount > 0$

D0201 [C] Données supplémentaires cantonales disponibles

hasMK

ACTION : hasMK=FALSE; V0304="0";

D1. Patients

D1000 [C] Nombre de cas (type A) : Sorties pendant la période relevée

strcmp(V0202,'A') == 0

D1001 [C] Nombre de cas (type B) : Cas entrés en cours d'année et toujours hospitalisés au 31.12

strcmp(V0202,'B') == 0 && strcmp(V1301,'3') == 0

D1002 [C] Nombre de cas (type C) : Séjour de longue durée

strcmp(V0202,'C') == 0 && strcmp(V1301,'3') == 0

D1010 [C] Nombre de sorties (cas A hospitalisés)

strcmp(V0202,'A') == 0 && strcmp(V1301,'3') == 0

D1011 [C] Cas ambulatoire

strcmp(V1301,'1') == 0

ACTION : doWriteRec=FALSE;

D1012 [C] Répartition des patients selon le type de prise en charge : Nombre de patients en semi-hospitalisation

strcmp(V0202,'A') == 0 && strcmp(V1301,'2') == 0

D1020 [C] Répartition des patients selon le sexe : Nombre de patientes

strcmp(V0202,'A') == 0 && strcmp(V1101,'2') == 0

D1030 [C] Répartition des patients selon l'âge (type A) : âge moyen

strcmp(V0202,'A') == 0

D2. Admission

D2008 [C] Séjour avant l'admission 8 autres (données non spécifiques)

strcmp(V0202,'A') == 0 && strcmp(V1202,'8') == 0

D2009 [C] Séjour avant l'admission 9 inconnu (données non spécifiques)

strcmp(V0202,'A') == 0 && strcmp(V1202,'9') == 0

D2018 [C] Mode d'admission 8 autres (données non spécifiques)

strcmp(V0202,'A') == 0 && strcmp(V1203,'8') == 0

D2019 [C] Mode d'admission 9 inconnu (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1203,'9') == 0`

D2028 [C] Décision d'envoi 8 autres (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1204,'8') == 0`

D2029 [C] Décision d'envoi 9 inconnu (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1204,'9') == 0`

D3. Séjour

D3009 [C] Classe 9 inconnue (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1302,'9') == 0`

D3018 [C] Prise en charge 8 autres (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1402,'8') == 0`

D3019 [C] Prise en charge 9 inconnue (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1402,'9') == 0`

D3029 [C] Type de prise en charge 9 inconnu (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1301,'9') == 0`

D3100 [C] Répartition selon la durée de séjour : Durée moyenne du séjour hospitalier

`strcmp(V0202,'A') == 0 && strcmp(V1301,'3') == 0`

D4. Sortie

D4005 [C] Décision de sortie : Cas de décès

`strcmp(V0202,'A') == 0 && strcmp(V1502,'5') == 0`

D4008 [C] Décision de sortie : 8 autres (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1502,'8') == 0`

D4009 [C] Décision de sortie : 9 inconnu (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1502,'9') == 0`

D4018 [C] Séjour après la sortie : 8 autres (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1503,'8') == 0`

D4019 [C] Séjour après la sortie : 9 inconnu (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1503,'9') == 0`

D4028 [C] Prise en charge après la sortie : 8 autres (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1504,'8') == 0`

D4029 [C] Prise en charge après la sortie : 9 inconnu (données non spécifiques)

`strcmp(V0202,'A') == 0 && strcmp(V1504,'9') == 0`

D5. Diagnostics

D5000 [C] Diagnostic principal : Donnée existante

`strcmp(MB->V0202,'A') == 0 && !DIA[0].isNull`

D5001 [C] Diagnostic : existant

`strcmp(MB->V0202,'A') == 0 && !DIA[i].isNull`

D5002 [C] Complément au diagnostic principal : Donnée existante

`strcmp(MB->V0202,'A') == 0 && !DIA[1].isNull`

D5003 [C] Diagnostic supplémentaire : Donnée existante

`i > 1 && strcmp(MB->V0202,'A') == 0 && !DIA[i].isNull`

D6. Traitements

D6000 [C] Traitement principal : Donnée existante

`strcmp(MB->V0202,'A') == 0 && !BEH[0].isNull`

D6001 [C] Traitement : existant

`strcmp(MB->V0202,'A') == 0 && !BEH[i].isNull`

Annexe A. Format des fichiers de métadonnées CIM/CHOP

Les tests des codes diagnostiques et des codes d'opération se font à l'aide de métadonnées. Il s'agit de deux tableaux, l'un pour la CIM-10, l'autre pour la CHOP.

La fonction INDEXCOL (code, col.) a été définie dans les tests de plausibilisation pour accéder aux colonnes de ces tableaux. Voir l'annexe C pour une description de ces fonctions.

CIM-10

1. Code	2. Type du code	3. Code compl.	4. Sexe	5. AgeMin	6. AgeMax
Liste de tous les codes CIM-10 à 3 et 4 positions existants qui sont utilisés pour les tests	0 = Causes externes / sans chap. Z 1 = Code + absolu avec * (1 :1) en titre 2 = Code * absolu (1 :n) 3 = Autre code conventionnel 4 = Code + absolu, sans * en titre 5 = Complément du chap. XX exigé 6 = Code CIM-10 à 3 positions non spécifique	Codes valables comme complément au code dague ou astérisque correspondant	Sexe autorisé avec le code, vide si le code n'est pas spécifique à un genre.	Limite d'âge inférieure admise avec le code	Limite d'âge supérieure admise avec le code

Les codes de type 6 peuvent en grande partie être remplacés par des codes à 4 positions valables, en rajoutant un .9 à la fin. Dans ces cas-là, le champ Code compl. donne le code à 4 positions correspondant.

CHOP :

1. Code	2. Type du code	3. Code compl.	4. Sexe	5. AgeMin
Liste de tous les codes CHOP à 2, 3 et 4 positions existants.	1 = non défini 2 = Code à 2 positions, non spécifique 3 = Code à 3 positions, non spécifique 4 = Code valable à 4 positions 5 = Code valable à 3 positions 6 = Code valable à 6 positions 7 = Code à 5 positions	Liste des codes complémentaires valables. Pas encore défini.	Sexe autorisé avec le code, vide si le code n'est pas spécifique à un genre. 1 = homme 2 = femme	Pas encore défini.

6. AgeMax	7. Diagnostic principal CIM-10	8. Valable à partir de	9. Valable jusqu'à	10. Code de rempl.
Pas encore défini.	Diagnostic principal qui correspond au traitement. Pas encore défini.	Année à partir de laquelle le code peut être utilisé.	Année jusqu'à la fin de laquelle le code peut être utilisé.	Code par lequel est remplacé un code non valable ou révisé.

Annexe B. Description des fonctions utilisées dans les tests

Les tests décrits dans les chapitres précédents font appel à des fonctions prédéfinies qui sont décrites ci-dessous. Remarquez que, d'une manière générale, les tests de MedPlaus ont une syntaxe orientée C++.

=
attribution var = valeur (is)
V0301='0';

==
correspond à, en comparaison (equal)
strlen(V0002) == 0

!=
ne correspond pas à, en comparaison (not equal)
strlen(V0008) != 0

<
plus petit que (less than)
atoi(V0008) < 3

>
plus grand que (greater than)
strlen(V0103) > 2

!
non (NOT)
!(dateV0004.isNull)

&&
et (AND)
strlen(V0009) != 0 && !isHex(V0009)

||
ou (OR)
!(dateV0004.isNull || dateV0004.isYYYYMMDD)

+
addition
(MB->STAYH+15)

*
multiplication
*(dateV1501.DateValue - dateV1201.DateValue + 1) * 24*

/
division
(atoi(V2302)/10) > 44

for ()
Boucle POUR. Exécute une instruction aussi longtemps que la condition dans la boucle POUR est remplie ++ compte de manière croissante

for (int di=0; di <=DIACount;di++)

->

Renvoie à l'enregistrement contenant la variable („Pointer“)

strcmp(MB->V0202,'A') == 0

atoi

Convertit les valeurs de la chaîne en valeurs entières (StringToInt)

isInt(V1103) §§ atoi(V1103)

atoi

Convertit les valeur de la chaîne en valeurs entières (StringToInt)

atoi(V0008) < 3

BEH[pos]

Fonctions relatives au traitement. Examine les variables concernant le traitement

pos indique la position (0 = TP, 1 = 1er TC, 2 = 2ème TC, 3 = 3ème TC etc.)

.isNull : Le champ destiné au traitement est vide

.Data[CHOP_] : Attributs du code

.Key : Contenu du champ

BEH[0].isNull

datevar

Fonctions relatives à la date. Examine les variables concernant la date

.isNull : Le champ destiné à la date est vide

.isYYYYMMDD : Le champ de la date possède le format AAAAMMJJ

.isYYYY0000 : Le champ de la date possède le format AAAA0000

.DateValue : Valeur de la date

.hasDate : Contient une date valable

.hasYear : Contient une année valable

.hasMonth : Contient un mois valable 1 – 12

.hasHour : Contient une heure valable 0 – 23

.hasMinute : contient un nombre de minutes valable 0 - 59

.Hour : valeur des heures

!(dateV0004.isNull)

DIA[pos]

Fonctions relatives au diagnostic. Examinent les variables concernant le diagnostic

pos fournit la position (0 = DP, 1 = SDP, 2 = 1er DS, 3 = 2ème DS etc.)

.isNull : Le champ destiné au diagnostic est vide

.Data[ICD_] : Attributs du code

.Key : contenu du champ

strlen(DIA[0].Key) > 0

DIACount

Compte le nombre de diagnostics dans un enregistrement

for (int di=0; di <=DIACount;di++)

doWriteRec

Décide si l'enregistrement doit être écrit

= FALSE (non)

= TRUE (oui)

doWriteRec=FALSE;

hasDiaLike('DIA')

Examine si un enregistrement contient le diagnostic commençant par 'DIA'

!hasDiaLike('Z38')

hasMN

Examine si un enregistrement contient un supplément Nouveau-nés
!hasMN

hasMP

Examine si un enregistrement contient un supplément psychiatrique
hasMP=FALSE

hasF64

Examine si un enregistrement contient un diagnostic F64
!hasF64

hasZ38_147

Examine si un enregistrement contient un diagnostic Z38
!hasZ38_147

BOOL hasKeyLikeLen(char* key, int len, char* keylist)

Examine si substr(key,0,len) se trouve sur la liste. Les éléments d'une liste sont séparés par des pipes.
hasKeyLikeLen(DIA[i].Key,3,'Z42|Z44|Z47|Z48|Z50|Z51|Z54')

isAlphaNumUpperCase(var)

Examine si la chaîne ne contient que des caractères relatifs à la quantité ['0'..'9','A'..'Z']
!isAlphaNumUpperCase(V4601)

isExklDia(DIA[pos].Key)

Examine l'existence d'une série de codes diagnostics spécifiques déterminés.
isExklDia(DIA[0].Key)

isHex(var)

Livre la valeur booléenne true lorsque la donnée var est hexadécimale
!isHex(V0009)

isIn(var,nr,_list)

Livre la valeur logique du résultat de la recherche de la valeur var dans une liste indexée (true : trouvée, false : non trouvée)
!isIn(V0202,3_ABC)

isInt(var)

Livre la valeur booléenne true si la donnée var est un nombre entier
isInt(V1103)

isMKCH

Examine si l'enregistrement est un enregistrement cantonal complémentaire MK
isMKCH $\mathcal{E}\mathcal{E}$ strlen(V5102) == 0

isSSDia(DIA[pos].Key)

Examine si le diagnostic est un diagnostic de grossesse : O03-O07 ; O10-O16 ; O20-O29 ; O30-O48 ; O60-O75 ; O80-O82 ; Z34-Z37
isSSDia(DIA[i].Key)

BOOL isSubKeyIn(char *str, int offset, int len, int size, char* array[])

SubKey = substr(str,offset,len). Livre la valeur VRAI lorsque la matrice indiquant la taille des enregistrements contient une valeur commençant par SubKey.
isSubKeyIn(DIA[i].Key,1,2,16,_16_DIA_PERINAT_LEN2)

LineNumberInMX

Compte le nombre d'enregistrements
LineNumberInMX > atol(MX->V0008)

STAYH

Durée du séjour (en heures)
(*MB->STAYH+15*)

char * strchr(char * str, int character)

Localise la 1ère apparition d'un caractère particulier dans une chaîne de caractères
strchr('0123456',V2302[2]) == NULL

strcmp(var,'value')

Livre le résultat d'une comparaison de 2 chaînes de caractères (StringCompare)
strcmp(V1104,'CHE') == 0

strncmp(var,'value',pos)

Livre le résultat d'une comparaison d'une position particulière de 2 chaînes de caractères (StringCompare)
strncmp(DIA[0].Key,'Z38',3) != 0

strlen(var)

Livre le nombre de caractères contenus dans var (StringLength)
strlen(V0002) == 0

var_list.inIndex

Examine si la valeur d'une variable est contenue dans une liste de l'index
!(V1104_regio.inIndex)

Remarque

ACTION

Les fonctions caractérisées par la dénomination ACTION ne sont exécutées que dans le mode spécial de correction propre à MedPlaus. Et ce, uniquement si la condition de test est remplie.

V0301='0';