

## Hoe werkt dat 'onder water': PDM2012 DataDictionary

### Wat is een DataDictionary?

Wikipedia omschrijft dit als volgt

*A **metadata repository** is a database created to store **metadata**. Metadata is information about the structures that contain the actual data. Metadata is often said to be "data about data", but this is misleading. Data profiles are an example of actual "data about data". Metadata is one layer of abstraction removed from this – it is data about the structures that contain data. Metadata may describe the structure of any data, of any subject, stored in any format.*

Met name de zinsnede 'information about the structures' dekt de lading als we over PD&M2012 praten. Echter, naast de documentatie van de structuren zijn bij PD&M2012 ook de verwijzingen naar de taaltteksten en veld-verificatie-richtlijnen opgenomen in de DataDictionary. Zowel voor lezen (alle overzichten) als schrijven (een aantal nieuwe modules) is de DataDictionary de spin in het PD&M2012-web. Ook de zoekschermen (PopUp box in het PD&M spraakgebruik) is onderdeel van de DataDictionary.

### Hoe ziet zo'n 'structure beschrijving' er uit ?

De files PD&Mfs1.\* op de ..\bdisk bevatten de specificaties van de bestandsindelingen. In deze bestandsindelingen worden de velden beschreven en staat de verwijzing naar taalfile nr en recordnr. voor de taalgebonden kolomtekst (idem rubriektekst).

Specificatie van een veld:

Type variabele {\$,%,!,#)		\$Char 1
Startpositie in recordbuffer		%Char 2 en 3
Lengte in recordbuffer		%Char 4 en 5
Link field		%Char 6 en 7
Index Tabel verwijzings nummer		%Char 8 en 9
Recordsoort 1=geldig		\$Char 10

Module nr waarin rubriektekst staat	Val Char 1 t/m. 3
Recordnr waarin rubriektekst staat	Val Char 4 t/m. 6

Toepassingsvoorbeeld van lezen lanfile nr en bijbehorend recordnummer uit variabele van DataDictionary DD\_LANFILE\_RECORD\$ waarbij t\_int% is de For/Next waarde in veldnummerreeks:

```
h4$ = Mid$(DD_LANFILE_RECORD$, 1 + SEGMENTNR_LNG& * 500 + (t_int% - 1) * 10, 10)
LANfile_LNG& = Val(Mid$(h4$, 1, 3))
LANRECORD_LNG& = Val(Mid$(h4$, 6, 3))
```

Voordeel van de numerieke benadering van datablokken is dat files (tabellen) als binary file geopend en bewerkt kunnen worden. Daarmee kunnen enorme snelheden bereikt worden onder Windows 10, ook in combinatie met het geïndexeerd zoeken zoals met 'Direct Search' in de lijsten plaatsvindt. Daarnaast kan het bestand via een Type/End Type structure als random access file benaderd worden.

### Voorbeeld van een type/end type structure

```
Type stock1_stm 'P101
  S1 As String * 10 'artikelcode
  S2 As String * 45
  S3 As String * 2
  etc
```

```
End Type
  Dim record10a As stock1_stm
```

```
Type stock1_ndx
  n1 As String * 10
  n2 As String * 2
```

```
End Type
  Dim record110a As stock1_ndx
```

```
FILESPECIFICATIONS PD&M --> STOCK1 .STM
primary RDBMS refers to artikelbestand dbl
Extended File Segment# 0
```

Field	type	start	check	length	total	link_ptr:	1st_Index#	Rec.Type	Phase	Error	Lan_File&Rec#
ARTCODE	1 \$	1	1	10	10	1	1	1	1	0	101_302è
BENAMING	2 \$	11	11	45	55	0	0	1	0	0	101 303i
EH	3 \$	56	56	2	57	0	55	1	0	0	101 903ò
INKOOP EH	4 \$	58	58	2	59	0	55	1	0	0	101 903ò
INKOOP EH2	5 \$	60	60	2	61	0	55	1	0	0	101 903ñ
FACTOR IE1	6 !	62	62	4	65	0	0	1	0	0	101R1954ò
FACTOR IE2	7 %	66	66	2	67	0	0	1	0	0	101R1955ò
KWALITEIT	8 \$	68	68	10	77	0	0	1	0	0	101 304i
NORM	9 \$	78	78	10	87	0	0	1	0	0	101 305i
VERR.PRIJS	10 !	88	88	4	91	0	0	1	0	0	101PF311Y
VERKOOPPR.	11 !	92	92	4	95	0	0	1	0	0	101R1312ò
VOORR.EIND	12 !	96	96	4	99	0	0	1	0	0	101PF365
BEW1	13 \$	100	100	4	103	0	6	1	0	0	101 956+
BEW2	14 \$	104	104	4	107	0	6	1	0	0	101 957ø
BEW3	15 \$	108	108	4	111	0	7	1	0	0	101 958ù
EXT.OMS. 1	16 \$	112	112	45	156	0	0	1	0	0	101 316ù
EXT.OMS. 2	17 \$	157	157	45	201	0	0	1	0	0	101 317ù
RECORD SRT	18 %	202	202	2	203	0	0	1	0	0	000PF000
TECH.VRD.	19 !	204	204	4	207	0	0	1	0	0	101PF375
RESERV.PR.	20 !	208	208	4	211	0	0	1	0	0	101PF376
IN BESTEL.	21 !	212	212	4	215	0	0	1	0	0	101PF379
EDIT OMS.	22 %	216	216	2	217	0	0	1	0	0	101R0366i
QTY/DOOS	23 %	218	218	3	220	0	0	1	0	0	101R3367ù
MAT.DIKTE	24 !	221	221	4	224	0	0	1	0	0	101R33240
OPP./EENH.	25 !	225	225	4	228	0	0	1	0	0	101R3325ù
KG/EENHEID	26 !	229	229	4	232	0	0	1	0	0	101R3326Ý
BESTELNIVO	27 !	233	233	4	236	0	0	1	0	0	101R3960è
GBREKENING	28 \$	237	237	10	246	0	0	1	0	0	101 323ò
VERK.PR. 2	29 !	247	247	4	250	0	0	1	0	0	101R1959ø
STUKKALK.	30 \$	251	251	4	254	0	0	1	0	0	101 329è
VERK.PR. 3	31 !	255	255	4	258	0	0	1	0	0	101R1967à
VALUTA/VP3	32 \$	259	259	4	262	0	0	1	0	0	101 968è
VERK.FACT.	33 !	263	263	4	266	0	0	1	0	0	101R1969à
ADV.BESTEL	34 \$	267	267	4	270	0	0	1	0	0	101 970à
LEVCODE1	35 \$	271	271	10	280	0	2	1	0	0	101 971ó
BESTELNR 1	36 \$	281	281	15	295	0	0	1	0	0	101 972ò
PTO-LEVCODE	37 \$	296	296	25	320	0	2	1	0	0	101 338ò
STUKLIJST	38 \$	321	321	8	328	0	30	1	0	0	101 335à
OUDE CODE	39 \$	329	329	15	343	0	0	1	0	0	101 973à
REKENDATUM	40 \$	344	344	17	360	0	0	1	0	0	000 000
LAATSTE B	41 \$	361	361	6	366	0	0	1	0	0	101 368ù
INTERNE BP	42 !	367	367	4	370	0	0	1	0	0	101PF369
FACT.PRIJS	43 !	371	371	4	374	0	0	1	0	0	101PF380
VAK-NUMMER	44 \$	375	375	6	380	0	0	1	0	0	101PF033
KORTING JN	45 \$	381	381	1	381	0	0	1	0	0	101\$0077Ñ
BTW-CODE	46 \$	382	382	1	382	0	0	1	0	0	101\$23700
IEH2!OUD+I	47 !	383	383	5	387	0	0	1	0	0	000??000
BESTELHOEV	48 !	388	388	4	391	0	0	1	0	0	101R1371á
MAX.VOORR.	49 !	392	392	4	395	0	0	1	0	0	101R1372æ
DIVERS/INT	50 \$	396	396	5	400	0	0	1	0	0	101 000

```
---> end file specs PD&M <---
```

In de PDM\_VB10library bevindt zich de onderstaande call `Load_DDIC_fieldinfo` voor het laden van gegevens uit de datadictionary. Om een lang verhaal kort te maken: dit stukje code beschrijft precies hoe de Data Dictionary gelezen wordt. Merk op dat de code redundant is voor het eerste segment ten opzichte van de andere acht segmenten.

```

Sub Load_DDIC_fieldinfo (TABNR_INT%, bdisk$, ldisk$, LAN$, RDBMSIndex$, DDIC_Fld$,
RDBMSFLD_SELECT$, Segment_nr_int%, QTY_fields$, reserve_3int%, reserve_1str$, reserve_2str$,
reserve_3str$)
ReDim startbytes%(50)

RDBMSFLD_SELECT$ = Charts_definitions$(TABNR_INT%, 1)
RDBMSIndex$ = Charts_definitions$(TABNR_INT%, 2)
Segment_nr_int% = Val(Charts_definitions$(TABNR_INT%, 3))

'* reset array welke naar DDIC velden en taal nrs kijkt MB25102013
For t1_ddic% = 1 To 100
    For t2_ddic% = 1 To 15 'cel parameter
        'cel parameter: 1= veldnr in ddic
        'cel parameter: 2= segmentnr in ddic
        'cel parameter: 3= lan nr gelezen uit ddic
        'cel parameter: 4= lan rechr gelezen uit ddic
        'cel parameter: 5= startbyte in record
        'cel parameter: 6= veldlengte in record
        'cel parameter: 7= tekensort
        'cel parameter: 8= range ondergrens
        'cel parameter: 9= range bovengrens
        'cel parameter: 10= koppeling via SOORT% aan tabelnr in boxspj / zie mcadP952 (CMD)
        'cel parameter: 11 = ingelezen LANtekst$
        'cel parameter: 12 = data uit record
        'cel parameter: 13 = upload_ptr is "1" als dit veld in naar file geschreven moet
        worden door ATforml_DDIC_write_filedata (reset naar "0" door
        ATforml_DDIC_get_filedata)
        'cel parameter: 14 = range_check
        'cel parameter: 15 = MEELIFTENDE, NIET WIJZIGBARE TEKST (BIJV. TELLERSTAND UREN p130)
        DDIC_Fld$(t1_ddic%, t2_ddic%) = "0"
    Next t2_ddic%
Next t1_ddic%

'* setting in datadictionary ophalen
Call load_SQL_datastructure(bdisk$, RDBMSIndex$, DD_BASIC_RECORD1$, DD_BASIC_RECORD2$,
DD_LANGUAGE_FILE_REC$, DD_extended_RECORD1$, DD_extended_RECORD2$)

'* tbv. debug info alles als string in de array gezet mb25102013
QTY_fields% = 0
Select Case Segment_nr_int%
    Case Is = 0 'basissegment

        Bytes_per_record& = 0
        For t% = 1 To 50
            '* fieldspec segment 1..8 ander record achter basisrecord, dus hanteer
            segment -1 in dit 4096 byte record
            '* startbyte
            FIELD_SETUP$ = Mid$(DD_BASIC_RECORD2$, 1 + (t% - 1) * 10, 10)
            '* veldlengte in byte bij $ (is 2 bij %, 4 bij ! en # bij #, resp
            16bitshort, single floatMBF en doublefloatMBF
            If Mid$(FIELD_SETUP$, 10, 1) = "1" Then 'test op rs=1
                Bytes_per_record& = Bytes_per_record& + CVI(Mid$(FIELD_SETUP$, 4, 2))
                'Print #9, t%, CVI(Mid$(FIELD_SETUP$, 4, 2)), Bytes_per_record&
            End If 'Mid$(FIELD_SETUP$, 10, 1) = "1" Then 'test op rs=1
        Next t% '1..50
        'Close #9
        For t% = 1 To Len(RDBMSFLD_SELECT$) Step 4
            fieldnumber$ = Mid$(RDBMSFLD_SELECT$, t%, 3)
            fieldnumber_lng& = Val(fieldnumber$)
            If fieldnumber_lng& >= 1 And fieldnumber_lng& <= 50 Then

                QTY_fields% = QTY_fields% + 1

                '* DDIC veldnummer (1..50) in DIT segment nr
                DDIC_Fld$(QTY_fields%, 1) = fieldnumber$

                '* segment nr (0...8)
                DDIC_Fld$(QTY_fields%, 2) = Str$(Segment_nr_int%)
            End If
        Next t%
    End Select

```

```

    '* basis + segment language in 1 record van 4500 bytes, dus hier met
       segment nr rekening houden
    '* lan file nummer
    DDIC_Fld$(QTY_fields%, 3) = Mid$(DD_LANGUAGE_FILE_REC$, 1 +
(Segment_nr_int% * 500) + (fieldnumber_lng& - 1) * 10, 3)

    '* lan record nummer
    DDIC_Fld$(QTY_fields%, 4) = Mid$(DD_LANGUAGE_FILE_REC$, 1 +
(Segment_nr_int% * 500) + (fieldnumber_lng& - 1) * 10 + 5, 3)

    '* range check LOAD_ddic_FIELDINFO
    'c:\demo\pdm\klad.vcs >> Range_check$ = UCase$(Mid$(DB_FldLanSpec$, 4, 2))
    Range_check$ = Mid$(DD_LANGUAGE_FILE_REC$, 1 + (Segment_nr_int% * 500) +
(fieldnumber_lng& - 1) * 10 + 3, 2)
    'R = numerieke range

    'm10032015 altijd terug naar tabnr%=1
    'en wijzigen velden blokkeren door PF=Protected field te initialiseren
    'voor ELK veld
    If Only_first_chart_view% = 1 Then 'm10032015
        Range_check$ = "PF"
    End If 'Only_first_chart_view% = 1 'm10032015

Select Case Range_check$
    Case Is = "PF"
        Result_Range_check$ = "PROTECTEDFIELD"
    Case Is = "R0"
        'FORMAT "..... 8L10 20L10
        Result_Range_check$ = "R 0 1"
    Case Is = "R1"
        Result_Range_check$ = "R 0 9999999"
    Case Is = "R2"
        Result_Range_check$ = "R -9999999 9999999"
    Case Is = "R3"
        Result_Range_check$ = "R 0 10000"
    Case Is = "R4"
        Result_Range_check$ = "R 0 18"
    'ORDERSTATUS
    Case Is = "$0"
        Result_Range_check$ = "$JN|Y=J" ' |Y=J
        betekent : Y wordt door J vervangen
    Case Is = "$1"
        Result_Range_check$ = "$J123"
    Case Is = "$2"
        Result_Range_check$ = "$012345"
End Select 'Range_check$
DDIC_Fld$(QTY_fields%, 14) = Result_Range_check$

    '* fieldspec segment 1..8 ander record achter basisrecord, dus hanteer
       segment -1 in dit 4096 byte record *****
    FIELD_SETUP$ = Mid$(DD_BASIC_RECORD2$, 1 + (fieldnumber_lng& - 1) * 10,10)
    '* startbyte
    DDIC_Fld$(QTY_fields%, 5) = Str$(CVI(Mid$(FIELD_SETUP$, 2, 2)))

    '* veldlengte in byte bij $ (is 2 bij %, 4 bij ! en # bij #, resp
       16bitshort, single floatMBF en doublefloatMBF
    DDIC_Fld$(QTY_fields%, 6) = Str$(CVI(Mid$(FIELD_SETUP$, 4, 2)))

    '* tekensoort reps. $#!#
    DDIC_Fld$(QTY_fields%, 7) = Mid$(FIELD_SETUP$, 1, 1)

    '* ondergrens (hier nog onbepaald)
    DDIC_Fld$(QTY_fields%, 8) = "0"

    '* bovengrens (hier nog onbepaald)
    DDIC_Fld$(QTY_fields%, 9) = "0"

    '* tabel nr (soort tbv. boxspv validatie)
    DDIC_Fld$(QTY_fields%, 10) = Str$(CVI(Mid$(FIELD_SETUP$, 6, 2)))

    '* language tekst ophalen en in array plakken
    DDIC_Fld$(QTY_fields%, 11) = String$(45, "?")

```

```

3) <= 999 Then
    If Val(DDIC_Fld$(QTY_fields%, 3)) > 0 And Val(DDIC_Fld$(QTY_fields%,
Close #94
Open ldisk + "mcdap" + Format$(Val(DDIC_Fld$(QTY_fields%, 3)), "000")
+ "." + LAN$ For Random As #94 Len = 80
    recordnummer& = Val(DDIC_Fld$(QTY_fields%, 4))
    If recordnummer& >= 1 Then
        Get #94, recordnummer&, record93a
        DDIC_Fld$(QTY_fields%, 11) = RTrim(record93a.lan1) +
String$(45, ".")
    Else
        MsgBox "missing lanrec# at DDIC_Fld$(QTY_fields%, 4) with
QTY_fields%=" + Str$(QTY_fields%) + " struc=" + RDBMSIndex$, vbOKOnly
    End If 'recordnummer& >= 1
Close #94
Else
    MsgBox "missing lanfile# at DDIC_Fld$(QTY_fields%, 3) with
QTY_fields%=" + Str$(QTY_fields%) + " struc=" + RDBMSIndex$, vbOKOnly
End If 'val(DDIC_Fld$(QTY_fields%, 3))>0 and
val(DDIC_Fld$(QTY_fields%, 3))<=999
End If 'fieldnumber_lng& >= 1 And fieldnumber_lng& <= 50 Then
Next t%

Case 1 To 8 'extra segment .X01 >>> .X08
'Close #9
'Open "c:\demo\pdm\test.txt" For Output As #9
'* bepalen recordlengte
Bytes_per_record& = 0
For t% = 1 To 50
    '* fieldspec segment 1..8 ander record achter basisrecord, dus hanteer
    segment -1 in dit 4096 byte record *****
    '* startbyte
    'MB11112013 FIELD_SETUP$ = Mid$(DD_extended_RECORD2$, 2 +
    ((Segment_nr_int% - 1) * 512) + (t% - 1) * 10, 10)
    FIELD_SETUP$ = Mid$(DD_extended_RECORD2$, 1 + ((Segment_nr_int% - 1) *
512) + (t% - 1) * 10, 10)
    '* veldlengte in byte bij $ (is 2 bij %, 4 bij ! en # bij #, resp
    16bitshort, single floatMBF en doublefloatMBF
    If Mid$(FIELD_SETUP$, 10, 1) = "1" Then 'test op rs=1
        Bytes_per_record& = Bytes_per_record& + CVI(Mid$(FIELD_SETUP$, 4, 2))
        'Print #9, t%, CVI(Mid$(FIELD_SETUP$, 4, 2)), Bytes_per_record&
    End If 'Mid$(FIELD_SETUP$, 10, 1) = "1" Then 'test op rs=1
Next t% '1..50
'Close #9

For t% = 1 To Len(RDBMSFLD_SELECT$) Step 4
    fieldnumber$ = Mid$(RDBMSFLD_SELECT$, t%, 3)
    fieldnumber_lng& = Val(fieldnumber$)
    If fieldnumber_lng& >= 1 And fieldnumber_lng& <= 50 Then

        QTY_fields% = QTY_fields% + 1

        '* DDIC veldnummer (1..50) in DIT segment nr
        DDIC_Fld$(QTY_fields%, 1) = fieldnumber$

        '* segment nr (0...8)
        DDIC_Fld$(QTY_fields%, 2) = Str$(Segment_nr_int%)

        '* basis + segment language in 1 record van 4500 bytes, dus hier met
        segment nr rekening houden * '* lan file nummer
        DDIC_Fld$(QTY_fields%, 3) = Mid$(DD_LANGUAGE_FILE_REC$, 1 +
(Segment_nr_int% * 500) + (fieldnumber_lng& - 1) * 10, 3)

        '* lan record nummer
        DDIC_Fld$(QTY_fields%, 4) = Mid$(DD_LANGUAGE_FILE_REC$, 1 +
(Segment_nr_int% * 500) + (fieldnumber_lng& - 1) * 10 + 5, 3)

        '>>mb06022014
        '* range check LOAD_ddic_FIELDINFO
        'c:\demo\pdm\klad.vcs >> Range_check$ = UCase$(Mid$(DB_FldLanSpec$, 4, 2))
        Range_check$ = Mid$(DD_LANGUAGE_FILE_REC$, 1 + (Segment_nr_int% * 500) +
(fieldnumber_lng& - 1) * 10 + 3, 2)
        'R = numerieke range
        Select Case Range_check$
            Case Is = "PF"

```

```

        Result_Range_check$ = "PROTECTEDFIELD"
    Case Is = "R0"
        'FORMAT
        Result_Range_check$ = "R      0      8L10      20L10"
    Case Is = "R1"
        Result_Range_check$ = "R      0      9999999"
    Case Is = "R2"
        Result_Range_check$ = "R     -9999999      9999999"
    Case Is = "R3"
        Result_Range_check$ = "R      0      10000"
    Case Is = "R4"
        Result_Range_check$ = "R      0      18"
    'ORDERSTATUS
    Case Is = "$0"
        Result_Range_check$ = "$JN|Y=J"
        betekent : Y wordt door J vervangen
    Case Is = "$1"
        Result_Range_check$ = "$J123"
    Case Is = "$2"
        Result_Range_check$ = "$012345"
End Select 'Range_check$

DDIC_Fld$(QTY_fields%, 14) = Result_Range_check$

'<<mb06022014

' * fieldspec segment 1..8 ander record achter basisrecord, dus hanteer
' segment -1 in dit 4096 byte record *****
FIELD_SETUP$ = Mid$(DD_extended_RECORD2$, 1 + ((Segment_nr_int% - 1) *
512) + (fieldnumber_lng% - 1) * 10, 10)

' * startbyte
DDIC_Fld$(QTY_fields%, 5) = Str$(CVI(Mid$(FIELD_SETUP$, 2, 2)))

' * veldlengte in byte bij $ (is 2 bij %, 4 bij ! en # bij #, resp
' 16bitshort, single floatMBF en doublefloatMBF
DDIC_Fld$(QTY_fields%, 6) = Str$(CVI(Mid$(FIELD_SETUP$, 4, 2)))

' * tekensoort reps. $!#
DDIC_Fld$(QTY_fields%, 7) = Mid$(FIELD_SETUP$, 1, 1)

' * ondergrens (hier nog onbepaald)
DDIC_Fld$(QTY_fields%, 8) = "0"

' * bovengrens (hier nog onbepaald)
DDIC_Fld$(QTY_fields%, 9) = "0"

' * tabel nr (soort tbv. boxspv validatie)
DDIC_Fld$(QTY_fields%, 10) = Str$(CVI(Mid$(FIELD_SETUP$, 6, 2)))

' * language tekst ophalen en in array plakken
DDIC_Fld$(QTY_fields%, 11) = String$(45, "?")
    If Val(DDIC_Fld$(QTY_fields%, 3)) > 0 And Val(DDIC_Fld$(QTY_fields%,
3)) <= 999 Then
        Close #94
        Open ldisk + "mcdap" + Format$(Val(DDIC_Fld$(QTY_fields%, 3)), "000")
+ "." + LAN$ For Random As #94 Len = 80
            recordnummer& = Val(DDIC_Fld$(QTY_fields%, 4))
            If recordnummer& >= 1 Then
                Get #94, recordnummer&, record93a
                DDIC_Fld$(QTY_fields%, 11) = record93a.lan1
            Else
                MsgBox "missing lanrec# at DDIC_Fld$(QTY_fields%, 4) with
QTY_fields%=" + Str$(QTY_fields%) + " struc=" + RDBMSIndex$, vbOKOnly
                End If 'recordnummer& >= 1
            Close #94
        Else
            MsgBox "missing lanfile# at DDIC_Fld$(QTY_fields%, 3) with
QTY_fields%=" + Str$(QTY_fields%) + " struc=" + RDBMSIndex$, vbOKOnly
            End If 'val(DDIC_Fld$(QTY_fields%, 3))>0 and
val(DDIC_Fld$(QTY_fields%, 3))<=999
            End If 'fieldnumber_lng% >= 1 And fieldnumber_lng% <= 50 Then
        Next t%

```

```
End Select 'Segment_nr_int%
```

```
DDIC_Fld$(QTY_fields% + 1, 11) = Format$(Bytes_per_record&, "00000000 ") + " <<  
Bytes_per_record& set by Load_DDIC_fieldinfo"  
DDIC_Fld$(QTY_fields% + 2, 11) = RDBMSIndex$ + " << RDBMSIndex$ set by Load_DDIC_fieldinfo"  
DDIC_Fld$(QTY_fields% + 3, 11) = Format$(Segment_nr_int%, "00000000 ") + " << Segment_nr_int%  
range[0..8] set by Load_DDIC_fieldinfo"
```

```
End Sub 'load_DDIC_fieldinfo
```