

IDA-BAZA NA DELTA/V V2.0 OPERACIJSKEM SISTEMU
DODATNA DOKUMENTACIJA PROGRAMSKEGA PROIZVODA
verzija 1.5

Iskra Delta Arhitektura - IDA BAZA

Iskra Delta -

FEBRUAR 1987

Koda:

VSEBINA

1.0	POPRAVKI NAPAK	3
1.1	Popravek "DBV.EXE"	3
1.3	Popravek "DBF.EXE"	3
2.0	PODPORA OPERACIJSKEGA SISTEMA DELTA V2.0.....	3
3.0	IZPOPOLNJEN PROGRAM DBVERY	3
4.0	NOV PROGRAM DDC	4

1 POPRAVKI NAPAK

1.1 Popravek "DBV.EXE"

a) Pri dodajanju, brisanju in relinku kodiranih podrejenih zapisov je v nekaterih primerih prišlo do poškodbe verige. Napaka je odkrita in popravljena.

b) V nekaterih primerih (podsheme, ki so imele READ ONLY dostop) je prihajalo do nenamernega zaklepanja podatkovnih zbirk tudi pri normalnem zaustavljanju jedra baze podatkov. Napaka je odkrita in popravljena.

1.2 Popravek "DBF.EXE"

Pri reformatiranju kombinirane zbirke zapisov je odpravljena lepotna napaka.

2 PODPORA OPERACIJSKEGA SISTEMA DELTA V2.0

IDA Baza V1.5 deluje na tudi na operacijskem sistemu VMS V4.2.

3 IZPOPOLNJEN PROGRAM DBVERIFY

Kljub vsem varnostnim ukrepom (logiranje, recovery) je možno, da pride do poškodb baze podatkov. DBVerify program odkriva napake v povezavah med zapisi, izgubljene zapise, konsistenco kombiniranih zbirk in izpiše protokol ugotovljenih poškodb ter statistično poročilo o pregledanih zbirkah. DBVerify se lahko izvaja v multiprogramskem okolju sočasno z drugimi programi, ki uporabljajo bazo podatkov. Priročnik za DBVerify se nahaja na vašem sistemu pod imenom

SYSSUPDATE:DBVERY.DOC. Program DBVERY ima v verziji 1.1 dodane nove funkcionalnosti: preverjanje po nadrejeni zbirki in preverjanje posameznih verig na osnovi ključa povezav. Odpravljene so napake pri analizi .SAV datotek (v glavnem v zvezi s kodiranimi zapisi).

4 NOV PROGRAM DDC

Iz razlogov, ki so opisani v Dodatni informaciji o programskih proizvodih IDA V1.5 je dodan nov program DDC (Data Definition Compiler). Navodilo za uporabo DDC se nahaja na vašem sistemu pod imenom SYSSUPDATE:DDC.DOC.


```
1 *****
2 *      DDC - Database Description Compiler V1.0      Ljubljana 1.1.1987 *
3 *****
4
5 *      1.      Compiling a Database Description
6
7 *      1.1.    Command Line Format
8
9 *      DDC [/qualifiers] file-specification
10
11 *      where:
12 *      DDC      - specifies the Database Description Compiler.
13 *      /qualifiers - specify compiler options.
14 *      file-spec. - specifies the files that contain the DDC sour
15 *                  descriptions. If you do not supply a file type
16 *                  in the file specification, the compiler assumes
17 *                  DDC as the default.
18
19 *      1.2.    Qualifiers
20
21 *      Qualifier      Default
22
23 *      /[NO]CONVERT      /NOCONVERT
24 *      /[NO]DESCRIPTOR  /DESCRIPTOR
25 *      /[NO]INIT        /NOINIT
26 *      /[NO]LOAD        /NOLOAD
27 *      /[NO]VALIDATE    /NOVALIDATE
28 *      /HELP
29
30 *      /CONVERT
31 *      /NOCONVERT
32 *      Converts DDP files to DDC files. Controls all DDP files:
33 *      Schema.SAV, Schema.EXT and through Schema.DBL controls
34 *      Run-time-schema.SAV and through Run-time-schema.DBL control
35 *      Subschema.SAV and Subschema.LIB. If any error occurs during
36 *      convert phase rewrite your Schema or Run-time-schema or
37 *      Subschema with DDP.
38 *      The default is /NOCONVERT.
39
40 *      /DESCRIPTOR
41 *      /NODESCRIPTOR
42 *      Controls whether the compiler produces an RUN-TIME-SCHEMA
43 *      and (or) SUBSCHEMA descriptor(s) files.
44 *      The default is /DESCRIPTOR.
45
46 *      /INIT
47 *      /NOINIT
48 *      Controls initialisation of LEX_SYSTEM files.
49 *      The default is /NOINIT.
50
51 *      /LOAD
52 *      /NOLOAD
53 *      Controls loading of LEX_SYSTEM after compile phase.
54 *      The default is /NOLOAD.
55
```



```
56 *      /VALIDATE
57 *      /NOVALIDATE
58 *      Set qualifiers /NOCOVERT/DESCRIPTOR/NOINIT/LOAD.
59 *      This qualifier is usable only with interactive DBDDP.
60 *      The default is /NOVALIDATE.
61
62 *      /HELP
63 *      Quick help facility.
64
65 *      1.3. Command Line Examples
66
67 *      DDC SCHEMA
68 *      Produces an object file DBV_SCHEMA:SCHEMA.001 and
69 *      listing file DBV_SCHEMA:SCHEMA.LIS from the source file
70 *      DBV_SCHEMA:SCHEMA.DDC. If source file consist
71 *      RUN-TIME-SCHEMA or SUBSCHEMA descriptions then produces
72 *      *.EXE file descriptors on directories DBV_PODROCJE or
73 *      DBV_PODSHEMA.
74
75 *      DDC/NODESCRIPTOR SCHEMA
76 *      Produces an object file DBV_SCHEMA:SCHEMA.001 and
77 *      listing file DBV_SCHEMA:SCHEMA.LIS from the source file
78 *      DBV_SCHEMA:SCHEMA.DDC.
79
80 *      DDC/CONVERT/NODESCRIPTOR SCHEMA000
81 *      Convert DBV_SCHEMA:SCHEMA000.SAV (DDP save files) and all
82 *      files with RUN-TIME-SCHEMA(s) and SUBSCHEMA(S) descriptions
83 *      of these SCHEMA000 into one DBV_SCHEMA:SCHEMA000.DDC file.
84 *      Produces an object file DBV_SCHEMA:SCHEMA000.001 and
85 *      listing file DBV_SCHEMA:SCHEMA000.LIS from the source file
86 *      DBV_SCHEMA:SCHEMA000.DDC.
87
88 *      DDC/LOAD/NODESCRIPTOR SCHEMA
89 *      Produces an object file DBV_SCHEMA:SCHEMA.001 and
90 *      listing file DBV_SCHEMA:SCHEMA.LIS from the source file
91 *      DBV_SCHEMA:SCHEMA.DDC.
92 *      Produces an sequential file DBV_SCHEMA:SCHEMA.002 with
93 *      entities and detected errors and load this file into
94 *      LEX_SYSTEM.
95
96 *      DDC/CONVERT/DESCRIPTOR/LOAD SCHEMA000
97 *      Convert DBV_SCHEMA:SCHEMA000.SAV (DDP save files) and all
98 *      files with RUN-TIME-SCHEMA(s) and SUBSCHEMA(S) descriptions
99 *      of these SCHEMA000 into one DBV_SCHEMA:SCHEMA000.DDC file.
100 *      Produces an object file DBV_SCHEMA:SCHEMA000.001 and
101 *      listing file DBV_SCHEMA:SCHEMA000.LIS from the source file
102 *      DBV_SCHEMA:SCHEMA000.DDC.
103 *      Produces an sequential file DBV_SCHEMA:SCHEMA000.002 with
104 *      entities and detected errors and load this file into
105 *      LEX_SYSTEM.
106
107 *      1.4. Compile-Time Error Messages
108
109 *      The format of the error message is:
110
111 *      *DDC-->F-Fatal error message
```



```
168
169 *          BLOCK contains 8 SECTORS
170 *          -----
171 *          BLOCK                = keyword
172 *          contains              = optional word
173 *          8                    = numeric literal
174 *          SECTOR               = condition name
175
176 *          3.  Schema Description Example
177
178 SCHEMA-DESCRIPTION
179
180     SCHEMA name is SALESA
181 *     -----
182     PASSWORD      is SALESA
183 *     -----
184 *     Schema is design of all database records. Each schema
185 *     is named, protected by password and contains all records
186 *     data names with item descriptions.
187
188     RECORD name is CSTOMR
189 *     -----
190 *     Record is collection of data items. For example, a record
191 *     of customer data would contain data items such as
192 *     "customer name", "address", "telephone number" and so on.
193
194     ITEM description is 05 OWNKEY PIC X(6)
195 *     -----
196 *     In every record, there is at least one data item that is
197 *     a "key". This element either identifies record unique or
198 *     as a member of a closely related group.
199 *     Length of item OWNKEY is 6 characters described with COBOL
200 *     picture "PIC" of the alphanumeric data field "X(6)".
201 *     "05" is level number of data item and represents the
202 *     hierachy of data in a record. You can use level numbers
203 *     from 05,06,07..... to 45.
204 *     You can use any of the following picture characters and
205 *     attributes (as used in COBOL).
206
207 *     Picture (PIC) Means
208 *     -----
209 *     A          Alphabetic data field
210 *     X          Alphanumeric data field
211 *     9          Numeric data field
212 *     S9         Signed numeric data field
213 *     9V9        Decimal point in numeric data field
214 *     9 CCMP     Numeric data field (binary format -integer)
215 *     9 CCMP-3   Numeric data field (packed decimal format)
216
217     ITEM description is 05 CSTINAM PIC X(50)
218     ITEM description is 05 ADRESS PIC X(60)
219     ITEM description is 05 TELEPH PIC 9(9)
220     ITEM description is 05 REMARK PIC X(6)
221
222     RECORD name is PRODUCT
223     ITEM description is 05     OWNKEY PIC X(12)
```



```
168
169 *          BLOCK contains 8 SECTORS
170 *          -----
171 *          BLOCK                = keyword
172 *          contains              = optional word
173 *          8                    = numeric literal
174 *          SECTOR               = condition name
175
176 *          3.  Schema Description Example
177
178 SCHEMA-DESCRIPTION
179
180     SCHEMA name is SALESA
181 *     -----
182     PASSWORD      is SALESA
183 *     -----
184 *     Schema is design of all database records. Each schema
185 *     is named, protected by password and contains all records
186 *     data names with item descriptions.
187
188     RECORD name is CSTQMR
189 *     -----
190 *     Record is collection of data items. For example, a record
191 *     of customer data would contain data items such as
192 *     "customer name", "address", "telephone number" and so on.
193
194     ITEM description is 05 OWNKEY PIC X(6)
195 *     -----
196 *     In every record, there is at least one data item that is
197 *     a "key". This element either identifies record unique or
198 *     as a member of a closely related group.
199 *     Length of item OWNKEY is 6 characters described with COBOL
200 *     picture "PIC" of the alphanumeric data field "X(6)".
201 *     "05" is level number of data item and represents the
202 *     hierachy of data in a record. You can use level numbers
203 *     from 05,06,07..... to 45.
204 *     You can use any of the following picture characters and
205 *     attributes (as used in COBOL).
206
207 *     Picture (PIC) Means
208 *     -----
209 *     A          Alphabetic data field
210 *     X          Alphanumeric data field
211 *     9          Numeric data field
212 *     S9         Signed numeric data field
213 *     9V9        Decimal point in numeric data field
214 *     9 COMP     Numeric data field (binary format -integer)
215 *     9 COMP-3   Numeric data field (packed decimal format)
216
217     ITEM description is 05 CSINAM PIC X(50)
218     ITEM description is 05 ADRESS PIC X(60)
219     ITEM description is 05 TELEPH PIC 9(9)
220     ITEM description is 05 REMARK PIC X(6)
221
222     RECORD name is PRDUCT
223     ITEM description is 05     OWNKEY PIC X(12)
```



```
224             ITEM description is 05      PRODUCT
225 * -----
226 * Item PRODUCT means all data of record PRODUCT
227
228             ITEM description is 11      PARTNO PIC X(60)
229             ITEM description is 11      PARTIAL REDEFINES PARTNO
230 * -----
231 * The REDEFINES clause allows different data description
232 * entries to describe the same storage area
233 * Item PARTIAL describe the same storage like item PARTNO
234
235             ITEM description is      22 PART01 PIC A(10)
236             ITEM description is      22 PART02 PIC X(20)
237             ITEM description is      22 PART03 PIC X(30)
238 * -----
239 * Item PARTIAL is also described with items PART01,
240 * PART02 and PART03.
241
242             ITEM description is 11      PRICE1 PIC 9(7)V99
243             ITEM description is 11      QUANTI PIC 9(7)V999
244
245 RECORD name is ORDERS
246 INDEX name is $A0000
247
248 * The record with defined INDEX name is treated to be
249 * COMBINED record. This is declared in Logical structure.
250
251             ITEM description is 05 OWNKEY PIC X(5)
252             ITEM description is 05 CUSINO PIC X(6)
253             ITEM description is 05 ALDATE OCCURS 2 times
254 * -----
255             ITEM description is 10 DATEAL PIC 9(6)
256 * -----
257 * OCCURS clause eliminates the need of separate entries for
258 * repeated data items. It defines tables and provides the
259 * basis for subscript and indexing.
260
261             ITEM description is 05 DATEOT REDEFINES ALDATE
262             ITEM description is 10 ORDDAT PIC 9(6)
263             ITEM description is 10 TRMDAT PIC 9(6)
264
265 * -----
266 * Record description without optional words !!!!!!!
267 RECORD ORDITM
268 ITEM 05 ORDERNO PIC X(5)
269 ITEM 05 DATAAL PIC X(28)
270 ITEM 05 DATAAL-05 REDEFINES DATAAL
271 ITEM 10 PARTNO PIC X(12)
272 ITEM 10 QUORDR PIC 9(5)V999
273 ITEM 10 QUDELI PIC 9(5)V999
274 * Record description without optional words !!!!!!!
275 * -----
276
277 RECORD name is SEQREC
278 ITEM description is 05      SEQDAT PIC X(120)
279
```



```
280          RECORD name is SYSREC
281          ITEM description is 05          OWNKEY PIC X(12)
282          ITEM description is 05          SYSDAT PIC X(108)
283
284  END-OF-DESCRIPTION * is the last statement in every description
```



```
DDC_VALIDATING      -- Fatais:  0 Warnings:  0 Informationals:  0
285
286 *      4.      Logical Structure Description Example
287
288 *      The logical structure of the shema is a description of all
289 *      owner-member relationship between record types. You can
290 *      have more then one relationship, each represent by its own
291 *      set. For any set you have to enter owner and member record
292 *      names, the record key (for the owner record), and the name
293 *      of the data item in the member record which is the set key.
294 *      - Owner record can be accesed by its ownkey and owns some
295 *      other records in relationship.
296 *      - Member or depended record not own any other record type.
297 *      - Combined records owns some other record type and is itself
298 *      owned by another record type.
299 *      - System record is owned only by system. This record can be
300 *      accesed by its ownkey and not owns any other record type.
301 *      There is only one logical structure description in schema.
302
303 LOGICAL-STRUCTURE-DESCRIPTION
304 LOGICAL-STRUCTURE name is SALESA
305
306     SET name is CSTORD
307             OWNER record name is CSTOMR
308             KEY item name is OWNKEY
309             MEMBER record name is ORDERS
310             KEY item name is CUSTNO
311     SET name is PRDORD
312             OWNER record name is PRDUCT
313             KEY item name is OWNKEY
314             MEMBER record name is ORDITM
315             KEY item name is PARTNO
316
317     SET name is $A0000
318 *
319 *      For combined record definition use INDEX name for
320 *      SET name. Memeber record and key must be declared
321 *      with NULL.
322             OWNER record name is ORDERS
323             KEY item name is OWNKEY
324             MEMBER record name is NULL
325             KEY item name is NULL
326
327 *-----
328 *      Set description without optional words !!!!!!!
329     SET ORDORD
330     OWNER ORDERS
331     KEY OWNKEY
332     MEMBER ORDITM
333     KEY ORDRNO
334 *      Set description without optional words !!!!!!!
335 *-----
336
337     SET name is SYSSET
338 *
339 *      For system record definition, memeber record and
```



```
340 *      key must be declared with NONE.  
341          OWNER record name is SYSREC  
342          KEY item name is OWNKEY  
343          MEMBER record name is NONE  
344          KEY item name is NONE  
345  
346 END-OF-DESCRIPTION
```


DDC_VALIDATING -- Fatais: 0 Warnings: 0 Informationals: 0

```
347
348 *      5.      Physical Structure Description Example
349
350 *      The physical structure description is description
351 *      of container files in a schema. There is only one
352 *      physical structure description in schema.
353
354 PHYSICAL-STRUCTURE-DESCRIPTION
355 PHYSICAL-STRUCTURE name is SALES
356 LOGICAL CONTAINER name is SALES
357 CONTAINER file name is DBV_SHEMA:SALES.CON
358
359 *      Container SALES has 4 different record descriptions
360 *      Record CSTOMR can occur 150 times in this container.
361
362 CONNECT record          CSTOMR
363 OCCURENCY number      is 150
364
365 *      Two different definitions of logical block length:
366 *      BLOCK contains          7  RECORDS
367 *      BLOCK contains          2  SECTORS
368 *      BLOCK contains          2  * (Records is default)
369
370 BLOCK contains          7  RECORDS
371 *
372 *      The block factor tells how many records can be placed
373 *      in a logical block. Length of logical block is 512,
374 *      1024, 1536, .... up to 8192 bytes.
375
376 CONNECT record          PRDUCT
377 OCCURENCY number      is 200
378 BLOCK contains          2  SECTORS
379 *
380 *      The block length is 2 sectors. Sector length is 512 btes.
381 *      The block length is 512 * 2 = 1024 bytes. Compiler will
382 *      calculate how many records can be placed in 2 sectors.
383
384 CONNECT record          ORDERS
385 OCCURENCY number      is 1000
386 BLOCK contains          3
387 CONNECT record          ORDITM
388 OCCURENCY number      is 2000
389 BLOCK contains          12  records
390 LOGICAL CONTAINER name is SYSREC
391 CONTAINER file name is DBV_SHEMA:SYSREC.CON
392 CONNECT record          SYSREC
393 OCCURENCY number      is 100
394 BLOCK contains          11  records
395 END-OF-DESCRIPTION
```


DDC_VALIDATING -- Fatal: 0 Warnings: 0 Informationals: 0

396
397 * 6. Run Time Schema Description Example
398

399 RUN-TIME-SCHEMA-DESCRIPTION
400 RUN-TIME-SCHEMA name is SALES1
401 PASSWORD is SALESA

402
403 ACTIVE programs is 10
404 *

405 * The maximum number of programs that can be simultaneously
406 * active. You can have 2 to 52 active programs. Default
407 * value is 10.

409 LOCKED records is 500
410 *

411 * The maximum number of records that can be locked by all
412 * programs at one time. This can be 0 to 9999. Default
413 * value is 300.

415 ACCESS time is 30
416 *

417 * The maximum time interval, in seconds, before a record
418 * that one program has locked becomes automatically accessible
419 * to another program. This can be 0 to 9999 seconds.
420 * Default value is 60 seconds

422 I/O-AREA name is IOCSTO
423 COPY number is 1
424 CONNECT record CSTCMR

425 *
426 * In run-time database uses i/o areas for input/output
427 * operations. You can connect one or more records in
428 * one i/o area. Record can be connected to only one
429 * i/o area. Number of copies of one i/o area can be
430 * up to 32. The total number of copies of all i/o areas
431 * for all records can't exceed 512. Default copy number
432 * is 2.

434 I/O-AREA name is IOPRDU
435 COPY number is 1
436 CONNECT record PRDUCT
437 I/O-AREA name is IOORDE
438 COPY number is 1
439 CONNECT record ORDERS
440 I/O-AREA name is IOORDI
441 COPY number is 1
442 CONNECT record ORDITM

443 END-OF-DESCRIPTION

DDC_VALIDATING -- Fatais: 0 Warnings: 0 Informationals: 0

```
444
445 *      7.  Subschema Description Example
446
447 SUBSCHEMA-DESCRIPTION
448 SUBSCHEMA name is SALESA101
449 PASSWORD      is SALESA
450 PROCESS name is DEMODB
451
452         ACCESS-RIGHTS  is UPDATE
453 *
454 *      Access rights for all subschema records is UPDATE or
455 *      READONLY. Default is READONLY. Default access is READONLY.
456
457         CONNECT subschema record CSTOMR001 from record CSTOMR
458 *
459 *      A subschema record is typically a part of schema record.
460 *      It contains only those data items from the schema record
461 *      that are needed by the program. You can define a record
462 *      more than one way for the same subschema. For example,
463 *      you can define the subschema record the first time so that
464 *      it has read and write access privileges; the second time,
465 *      perhaps only read access privileges.
466
467         RECORD-PROTECTION  is SHARED
468 *
469 *      Record protection is SHARED for all programs or PRIVILEGED
470 *      exclusive use only by one program. Default record protection
471 *      is SHARED
472
473         RECORD-ACCESS  is GETP GET
474 *
475 *      Subschema record can be accessed by:
476 *      GETP      - Read record by physical sequence
477 *      GET       - Read record with key
478 *      ADD      - Add or insert record
479 *      DEL      - Delete record
480 *      RWR     - Rewrite record
481 *      You can select as many as necessary. Default record access
482 *      is GETP.
483
484         SELECT item      OWNKEY  *PIC X(6)
485         SELECT item      CSTNAM  *PIC X(50)
486         SELECT item      ADRESS  *PIC X(60)
487         SELECT item      TELEPH  *PIC 9(9)
488         SELECT item      REMARK  *PIC X(6)
489 CONNECT subschema record CSTOMR002 from record CSTOMR
490 RECORD-PROTECTION  is SHARED
491 RECORD-ACCESS  is GET  INS  RWR
492         SELECT item      OWNKEY  *PIC X(6)
493         SELECT item      CSTNAM  *PIC X(50)
494         SELECT item      ADRESS  *PIC X(60)
495         SELECT item      TELEPH  *PIC 9(9)
496         SELECT item      REMARK  *PIC X(6)
497 CONNECT subschema record PRODUCT001 from record PRODUCT
498 RECORD-PROTECTION  is SHARED
```



```
499 RECORD-ACCESS is GETP GET
500           SELECT item      OWNKEY  *PIC X(12)
501           SELECT item      PRICE1  *PIC 9(7)V9(2)
502           SELECT item      QUANTI  *PIC 9(7)V9(3)
503 CONNECT subschema record PRODUCT002 from record PRODUCT
504 RECORD-PROTECTION is SHARED
505 RECORD-ACCESS is GET  INS  RWR
506           SELECT item      OWNKEY  *PIC X(12)
507           SELECT item      PARTNO  *PIC X(60)
508           SELECT item      PRICE1  *PIC 9(7)V9(2)
509           SELECT item      QUANTI  *PIC 9(7)V9(3)
510 CONNECT subschema record ORDERS001 from record ORDERS
511 RECORD-PROTECTION is SHARED
512 RECORD-ACCESS is GETP GET
513           SELECT item      OWNKEY  *PIC X(5)
514           SELECT item      CUSTNO  *PIC X(6)
515           SELECT item      ORDDAT  *PIC 9(6)
516           SELECT item      TRMDAT  *PIC 9(6)
517 CONNECT subschema record ORDERS002 from record ORDERS
518 RECORD-PROTECTION is SHARED
519 RECORD-ACCESS is GET  INS  RWR
520           SELECT item      OWNKEY  *PIC X(5)
521           SELECT item      CUSTNO  *PIC X(6)
522           SELECT item      ORDDAT  *PIC 9(6)
523           SELECT item      TRMDAT  *PIC 9(6)
524 CONNECT subschema record ORDERS003 from record ORDERS
525 RECORD-PROTECTION is SHARED
526 RECORD-ACCESS is GET  DEL
527           SELECT item      OWNKEY  *PIC X(5)
528           SELECT item      CUSTNO  *PIC X(6)
529           SELECT item      ORDDAT  *PIC 9(6)
530           SELECT item      TRMDAT  *PIC 9(6)
531 CONNECT subschema record ORDITM001 from record ORDITM
532 RECORD-PROTECTION is SHARED
533 RECORD-ACCESS is GETP GET
534           SELECT item      ORDERNO *PIC X(5)
535           SELECT item      DATAAL  *PIC X(28)
536 CONNECT subschema record ORDITM002 from record ORDITM
537 RECORD-PROTECTION is SHARED
538 RECORD-ACCESS is GET  INS  RWR
539           SELECT item      ORDERNO *PIC X(5)
540           SELECT item      DATAAL  *PIC X(28)
541 CONNECT subschema record ORDITM003 from record ORDITM
542 RECORD-PROTECTION is SHARED
543 RECORD-ACCESS is GET  RWR
544           SELECT item      ORDERNO *PIC X(5)
545           SELECT item      PARTNO  *PIC X(12)
546           SELECT item      QUDELI  *PIC 9(5)V9(3)
547 CONNECT subschema record ORDITM004 from record ORDITM
548 RECORD-PROTECTION is SHARED
549 RECORD-ACCESS is GET  DEL
550           SELECT item      ORDERNO *PIC X(5)
551           SELECT item      DATAAL  *PIC X(28)
552 END-OF-DESCRIPTION
```



```
DDC_VALIDATING      -- FataIs:  0 Warnings:  0 Informationals:  0
553
554 *      8.      Subschema Logical Description Example
555
556 SUBSCHEMA-LOGICAL-DESCRIPTION
557 SUBSCHEMA name is SALESA101
558
559      ACCESS subschema record ORDERS002 with set CSTORD
560 *      -----
561 *      To enable subschema access to subschema record you
562 *      should specify the set you are going to use.
563
564 ACCESS subschema record ORDITM001 with set PRDORD
565 ACCESS subschema record ORDITM002 with set ORDORD
566 ACCESS subschema record ORDITM003 with set ORDORD
567 *      -----
568 *      Description without optional words !!!!!!!
569      ACCESS ORDITM004 ORDORD
570 *      -----
571 END-OF-DESCRIPTION
```


DDC VALIDATING -- FATALS: 0 WARNINGS: 0 INFORMATIONALS: 0

572

573 * 9. Sequential Record Description Example

574

575 SEQUENTIAL-RECORD-DESCRIPTION

576 SUBSCHEMA name is SALES101

577 SEQUENTIAL-RECORD name is SEQ001 from record SEQREC

578 RECORD-LENGTH is 120

579 *

580 * Record length is number between 14 and 510. Default is 14.

581

582 ACCESS-MODE is OUT

583 *

584 * IN Reading is permitted

585 * OUT Writing is permitted

586 * IN-OUT Reading and writing is permitted

587 * EXTEND Extending is permitted

588

589 ACCESS-PROTECTION is SHARED

590 *

591 * SHARED Simultaneous access to several program

592 * NOSHARED Exclusive access to one program

593

594 CR-LF characteristic is YES

595 *

596 * YES Record has carriage return control

597 * NO Record has not carriage return control

598

599 FILE name is DBV_SHEMA:SEQREC.SEQ

600

SELECT ITEM NONAME

*DDC->W-Default statement is inserted

*DDC->I-Statement is not implemented

601 END-OF-DESCRIPTION

DDC_VALIDATING -- Fatafs: 0 Warnings: 1 Informationals: 1

602

603

604 *

TABLE OF CONTENTS

605

606 * 1. Compiling a Database Description

607 * 1.1. Command Line Format

608 * 1.2. Qualifiers

609 * 1.3. Command Line Examples

610 * 1.4. Compile-Time Error Messages

611 * 2. DDC Language Elements

612 * 2.1. General Description

613 * 2.2. Statement Description

614 * 2.3. Statement Line Examples

615 * 3. Schema Description Example

616 * 4. Logical Structure Description Example

617 * 5. Physical Structure Description Example

618 * 6. Run Time Schema Description Example

619 * 7. Subschema Description Example

620 * 8. Subschema Logical Description Example

621 * 9. Sequential Record Description Example

622

623

624 *****

625 * DDC - Database Description Compiler V1.0 Ljubljana 1.1.1987 *

626 * Dusan Fiser *

627 *****

DDC_VALIDATION_END -- Fatafs: 0 Warnings: 1 Informationals: 1