![SWT Software Tools logo]

English User Manual

Global 2000

22/11/01 / 2022-09-01 008.384

Contents

1. Support

Whilst this product is supported under the normal TIS Software terms and conditions, a sufficient level of knowledge in the importing applications is required. The Global support department will only deal with problems with the TRIO software, and do not provide support for the host products (Microsoft Excel etc.).

1.1. Scope

The system supports the following Global-2000 filetypes stored in .SVL files:

RSAM   Relative Sequential (Non-Indexed) files
ISAM   Indexed files (1 index only)
DMAM   Indexed files

2. Setup

After TRIO and the ODBC driver have been installed you must setup the following:
>   **- A subsystem containing the Data dictionary for Global**
>   **- A driver for the Global-2000 file system**

The setup is made using the 'Data Dictionary', which can be started from the program group SWTOOLS.

2.1. Creating a new subsystem

The subsystem is created by selecting the following function from the menu 'File':
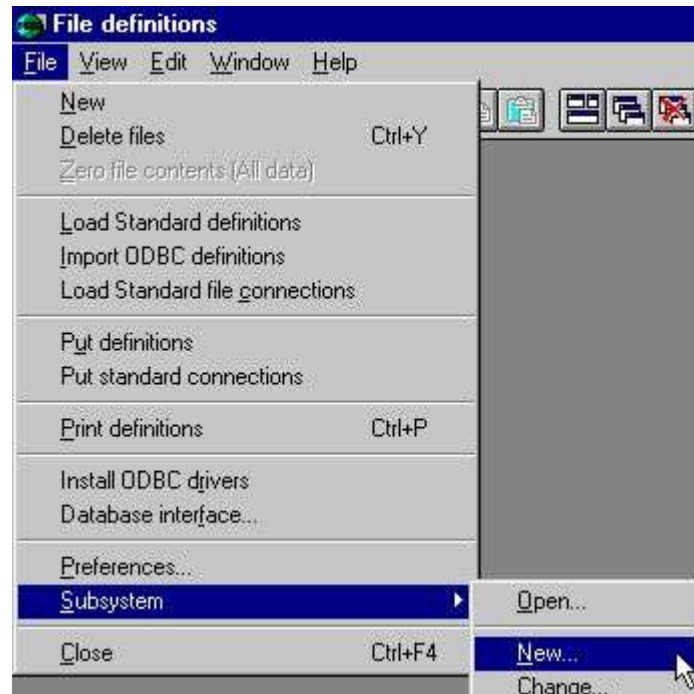
1. Creating a new subsystem

The most important information for the subsystem is the fields

> **- file definitions**
> **- reports**

These fields must be a path, local or on a network, where the data dictionary can be saved. If a non-existent path is entered it will automatically be created.

2. Important information on the subsystem

The field 'name' is used to give the system a logical and understandable name. The 'description' is free text. It will only be displayed when the mouse cursor is moved over the system name in the other TRIO applications. You may want to set up several subsystems for example one for live data and one for a test database.

2.2. Installing the driver for Global-2000

The driver may be installed using the application 'Data Dictionary'. The function to select is on the menu 'File':

3. Installing the Global-2000 driver

The function consists of a dialog window. This initially displays information on the driver type 'SSV Textfile'. To change the installed Global-2000 interface, select the type 'Global 2000' from the first combo box.
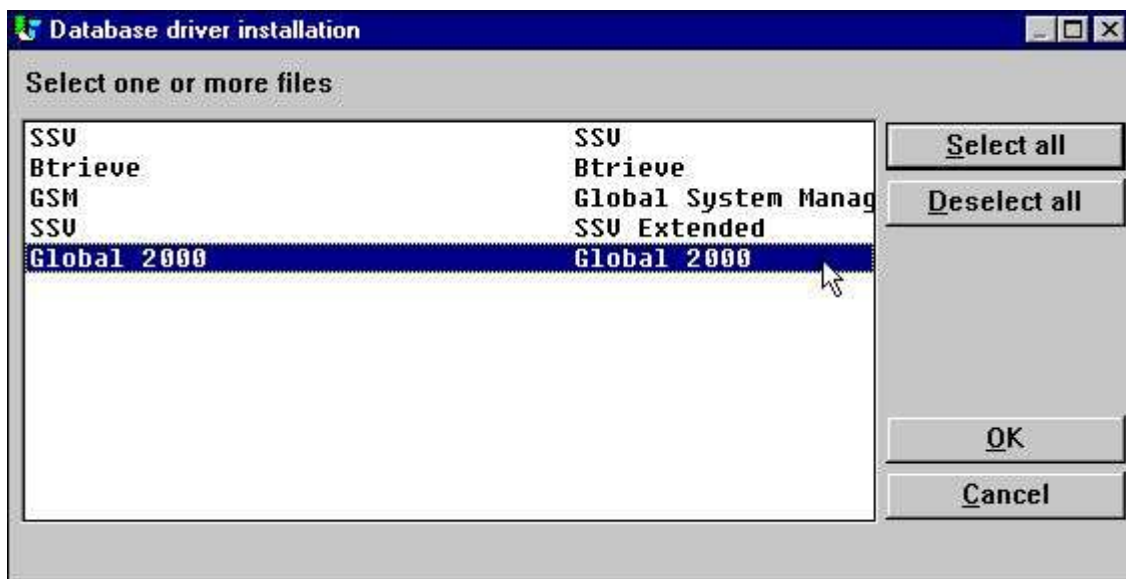
4. Selecting the driver for Global-2000

Select the Global-2000 driver and click OK.

Global 2000

The application will restart itself, and the Global-2000 file system will be installed. However, it is necessary for the driver to know which database to access on which unit.

2.3. Global-2000 interface adjustments

The database interface (driver) can be changed from the menu 'File' by selecting the function:

5. Changing the Global-2000 interface

2.3.1. Initial changes to the interface

Start by changing the 'Description' to a logical and understandable name. This allows the interface to be installed multiple times for different databases in the Global system.
Set the 'Read only' to 'Yes.'. The driver only supports reading the Global-2000 files - NO updates.
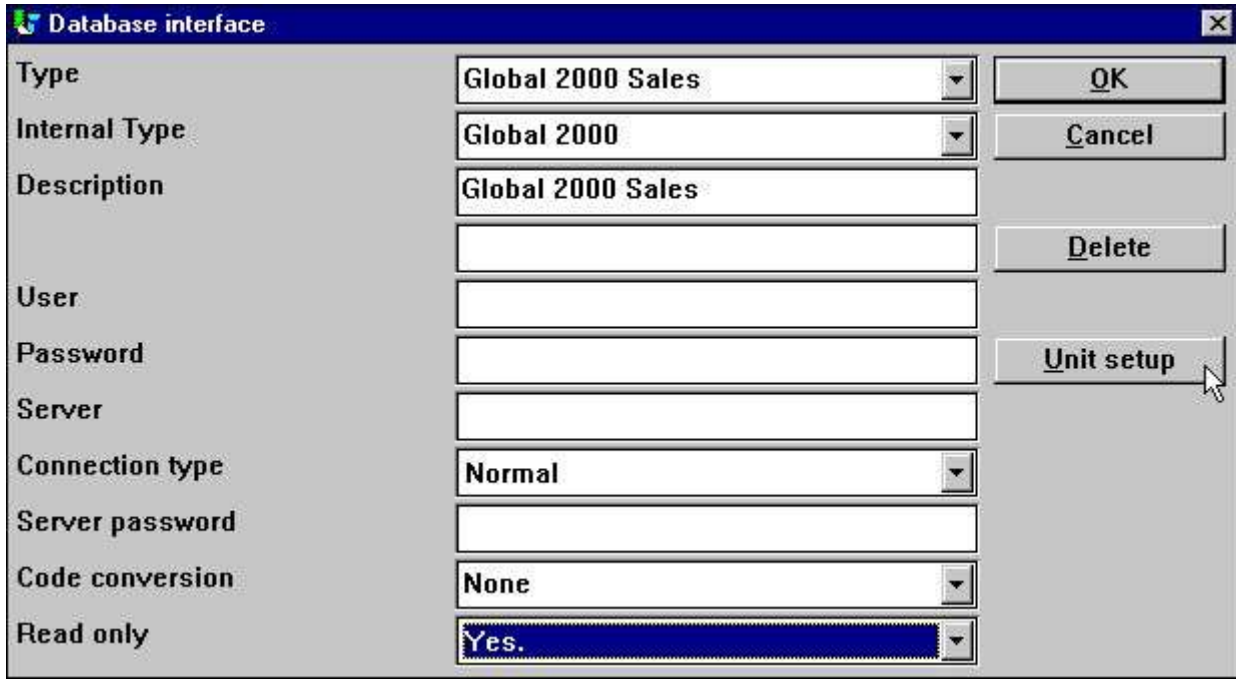
6. Initial interface changes

2.3.2. Creating the unit table

Now select the button 'Unit Setup'. This will activate a new dialog with the actual database information for the Global system.
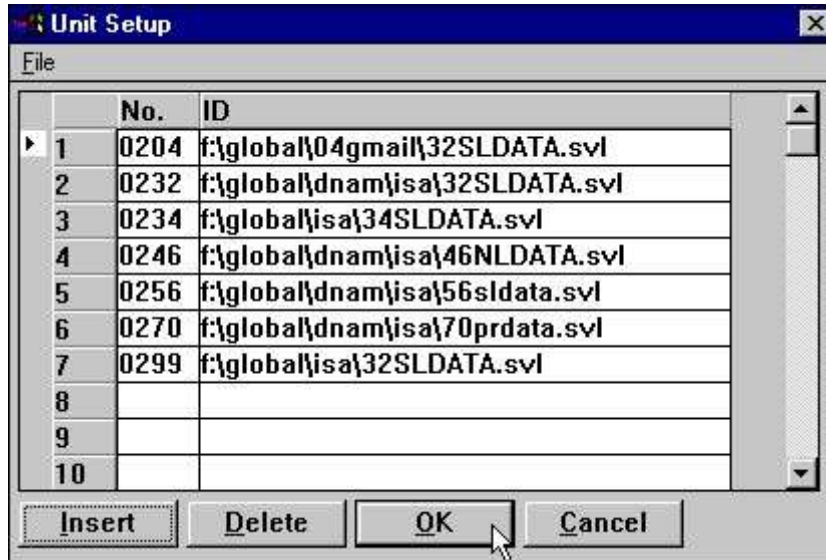
7. Setting up the unit table

A UNIT table must be created connecting units to actual SVL files. This unit table is placed together with the BASIS.SSV file, named as eg.

c:\swtools\unit0002.ssv

where 0002 is the driver number from basis.ssv. The file contains lines like

0234;i:\globalcd\g2000\34SLDATA.SVL

assigning unit 234 to the stated SVL file.

Please remember to enter the unit number as 4 digits, e.g. 234 must be entered as 0234.

2.3.2.1. Changing units and the unit assignment file

You may at any time change the unit assignment file moving complete units to another SVL file. You may also move individual files to another unit by changing the physical filename.

2.3.2.2. Use of the unit-table during loading of copybooks

Note that you should setup a correct unit table before loading the copybooks as filename and index informations are taken from the real files during this.

You may of course enter these informations manually or amend the unit table and reload the definitions if required.

3. Import of the copybooks to SW-Tools Dictionary

The Global-2000 system has no own data dictionary which means the only way to get the file definitions is from the COPYBOOK used for system generation.
The copybook must be placed beforehand in textform on the same path as where the SW-Tools file definitions are to be stored (DMF path) and named as G2000.TXT, eg.

c:\swtools\dmf\g2000.txt

Due to the varying structure of the copybooks you may find it convenient to make minor adjustments to the G2000.TXT file before loading. Several copybooks may also be combined into this file for loading in one go.
The import may then be performed from the following menu:
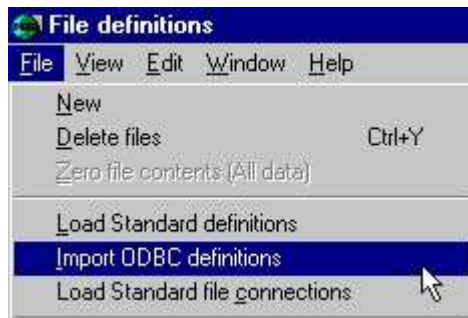
8. Importing a copybook from the Global-2000 system to the Data-Dictionary

Before the file descriptions for the database can be imported, the newly installed interface must be selected. This is done in the upper right corner where the function normally has the selection 'SSV Textfile'.
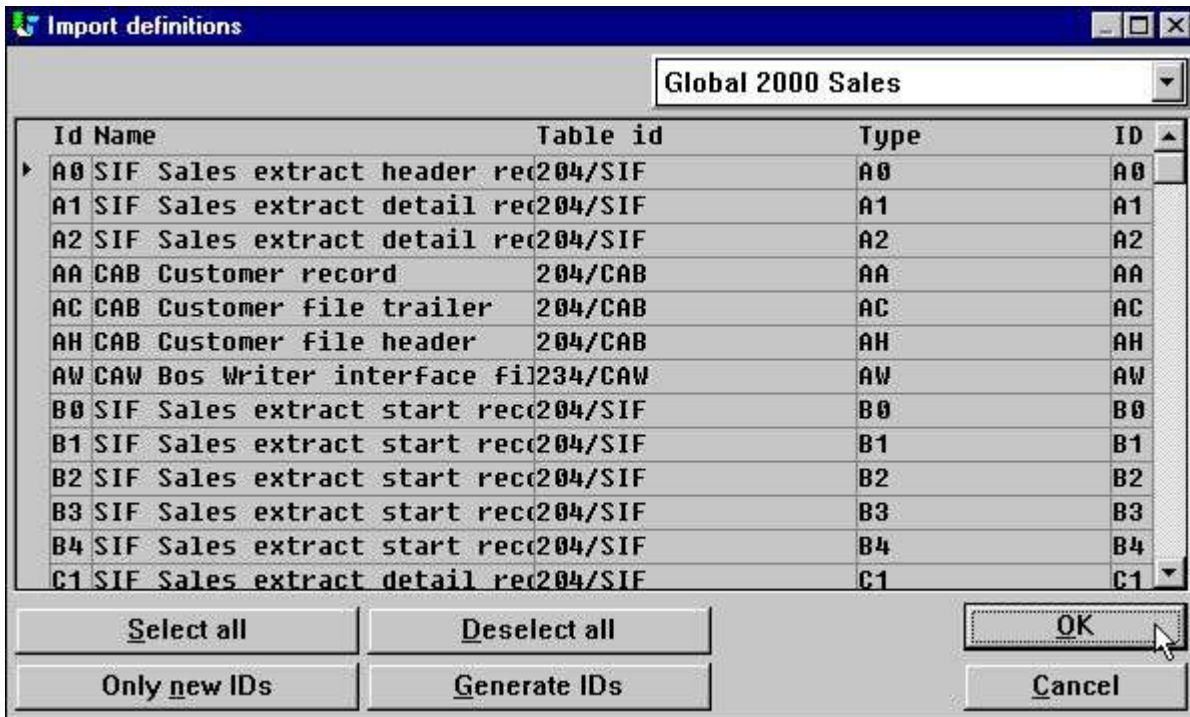
9. List of files in database

The files will by default be imported with the same file id's as known in the Global-2000 system, e.g. recordtype VP for Sales analysis detail records.
By selecting 'OK' all file descriptions will be imported.
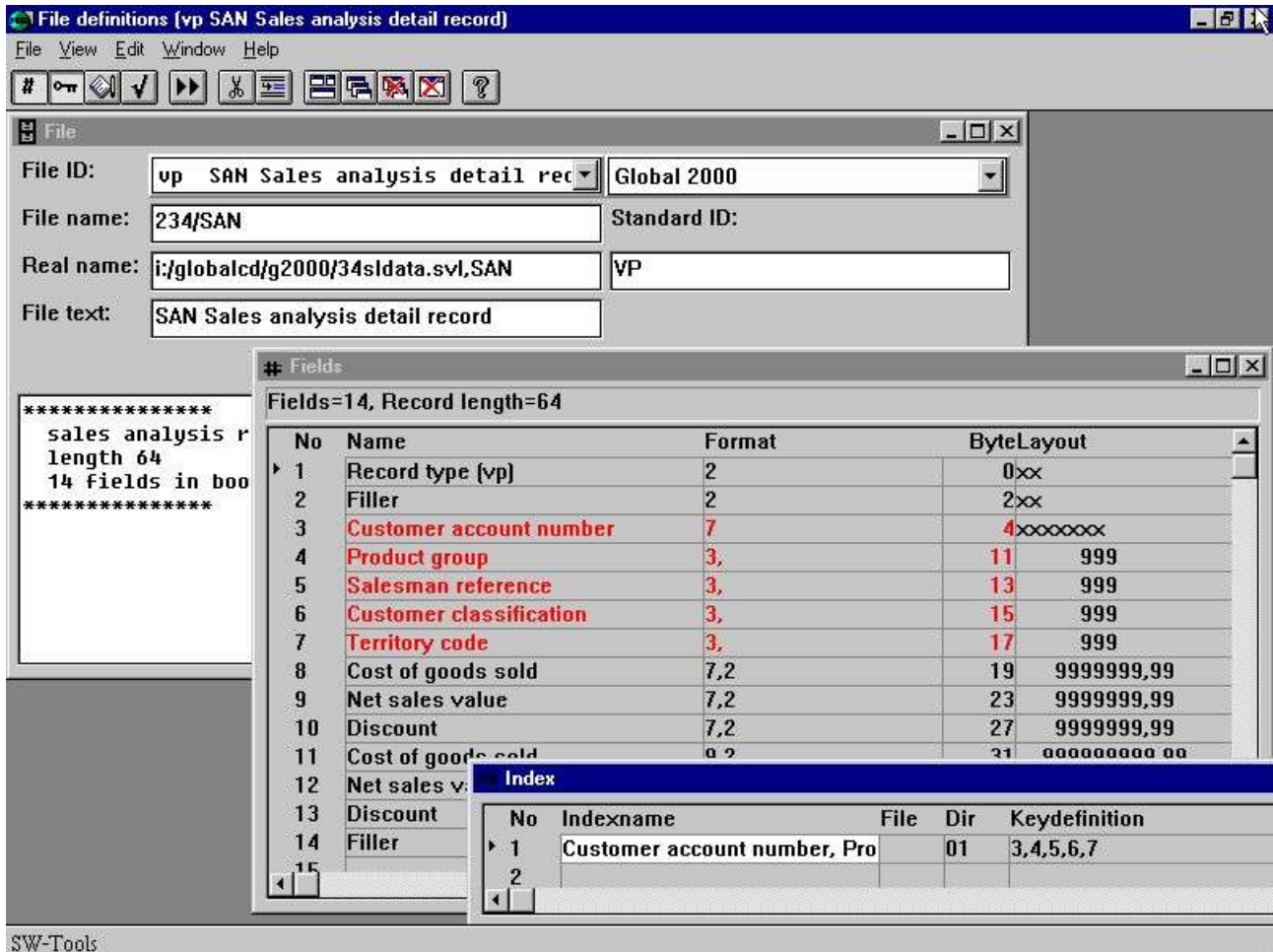
10. Example of a loaded table

3.1. Filename

During the load the files will be searched in all SVL files mentioned in the unit table and assinged a unit from here. The CAB file may be found on unit 234 whereas the filename will be loaded as 234/CAB.

3.2. Standard ID

The STANDARD ID will be set to the recordtype of the file and will be used to select only records of this type by read. You should note that where you may choose the file ID freely then the standard ID will have to correspond to the real recordtype stored in the file.

3.3. Keyfields

The SVL file header will be read during import of definitions while the unit table must have been setup before this.

The unitnumber is inserted for the file as 234/CAB and the keylength is used to define all fields as keys when these has a byte displacement between 4 and 4+keylength (byte 0-3 are reserved for recordtype and link always).

3.3.1. RSAM file indexes

As the RSAM relative sequential files have no index these will cause a problem when working with Microsoft ACCESS as an unique key is required here.

Due to this an extra field will be added to a RSAM file definition as

RECORDNUMBER 8,W

and this field will be setup as index keyfield fooling ACCESS to believe an index is present.

3.3.2. DMAM file indexes

Due to the structure used in DMAM the following index options will be common in the key definetions:

```
D  = Descending index, all bits in the field are inverted
S  = No inversion on sign bit in numeric field
     Set when more numeric fields are included into one DMAM keypart
     as DMAM will flip the signbit on the first field only.
Cx = Compression using translation table
     x is stated as the DMAM used length of the compressed key
M  = Multiple keys, set as documentation only for non-unique keys
```

.

3.3.3. DMAM file indexes and the ODBC driver

The ODBC knows which index to use from the WHERE clause in the SQL, the ANSI SQL standard specifies how this should be done and you may find some samples also in the README textfile on the ODBC driver disk.

WHERE Salesman>1

will go thru all defined index of the file looking for an index with the first field being 'Salesman' using this if present.

If no such index is found the first index is used and the file may be read in complete.

3.4. Field packtypes

The same packtypes as for Global 3000 are used by default:

1112 PT_GSM1 x bytes, Alphanumeric
1113 PT_GSM2 1 byte, Integer <=2 digits
1114 PT_GSM3 2 bytes, Integer <=4 digits
1115 PT_GSM4 3 bytes, Integer <=6 digits
1116 PT_GSM5 4 bytes, Integer <=9 digits
1117 PT_GSM6 1-8 bytes, Floating point if decimals or more than 9 digits

4. Scanning of the copybook

You must be fully aware that the loading of definitions from the copybooks should be considered as guidelines for the file definitions only and should not be taken without control as the loading can never be better than the original copybook.
Amendments of the file definitions after loading will be usual as the copybooks does not state all details and does not always follow any fixed syntax. It may be more convenient to amend the G2000.TXT copy of the copybook before loading to make this follow a more general syntax.

4.1. Multiple copybooks

In case of multiple copybooks to be loaded into the same FDF system you may select to join these into the G2000.TXT file. By import of the 2.copybook the fileID's cannot be the reused and the defaulting to the file recordtypes cannot always be done.

You may also install the Global 2000 database interface multiple times in the same system. If you change the 'Description' field when 'Amending database interface' to something other than 'Global 2000' next 'Install odbc drivers' will give you a second instance.

Each of these has there own unit table, say UNIT0002.SSV for the first UNIT0003.SSV for the second. By 'Import odbc definitions' you select the appropriate driver for the files giving you the associated unit table.

The copybook G2000.TXT must be overwritten between each such import.

4.2. A copybook example

A file definition from the Global-2000 copybook may look like:

```
:
RECORD VP        SAN Sales analysis detail record
:
.BOOK VP
****************
*   SALES ANALYSIS RECORD
*   LENGTH 64              KEY 15
*   14 FIELDS IN BOOK
****************
    02  &&TYPE  PIC X(2)                 * RECORD TYPE (VP)
    02  FILLER  PIC X(2)
    02  &&ACNO  PIC X(7)                 * CUSTOMER ACCOUNT NUMBER
    02  &&PGRP  PIC 9(3)        COMP    * PRODUCT GROUP
    02  &&SMAN  PIC 9(3)        COMP    * SALESMAN REFERENCE
    02  &&CUCL  PIC 9(3)        COMP    * CUSTOMER CLASSIFICATION
    02  &&TYCD  PIC 9(3)        COMP    * TERRITORY CODE
              *
              *   SALES STATISTICS FOR THE PERIOD
              *
    02  &&PCGS  PIC 9(7,2)      COMP    * COST OF GOODS SOLD
    02  &&PNET  PIC 9(7,2)      COMP    * NET SALES VALUE
    02  &&PDIS  PIC 9(7,2)      COMP    * DISCOUNT
              *
              *   SALES STATISTICS YEAR TO DATE
              *
    02  &&YCGS  PIC 9(9,2)      COMP    * COST OF GOODS SOLD
    02  &&YNET  PIC 9(9,2)      COMP    * NET SALES VALUE
    02  &&YDIS  PIC 9(9,2)      COMP    * DISCOUNT
    02  FILLER  PIC X(18)
.END
:
```

4.3. Filename

The filename is taken from the RECORD sentense, in this case SAN:

RECORD VP **SAN Sales analysis detail record**

if just a BOOK section is present as

.BOOK AA *** File CAB SL customer record (IS)**
:
:
.END

the file definition will be loaded but the filename cannot be taken from the comment here and Unit-information and indexdefinitions will not be supplied.

4.4. Filetext

The filetext is taken from the RECORD sentense also
> **RECORD VP   SAN Sales analysis detail record**

and if no RECORD sentense is present from the comment on the .BOOK line.

4.5. Record type

All files will have a record type which are taken from the comment line of the first field named TYP or RTYP

02 &&TYPE PIC X(2)   * RECORD TYPE (VP)
02 &&RTYP PIC X(2)   * RECORD TYPE = DH

A field named 'Record type (VP)' is defined as a 2 alphanumeric field and the STANDARD ID for the file is set to VP causing all read to check this type.

The first &&TYP or &&RTYP is taken, record type is set to (xx) or to the two last alphanumeric characters on the line.

If no recordtype can be found this way the file shortname from the .BOOK sentense will be used.

4.6. Numeric fields

Numeric fields are stored in different length. Depending on the COBOL PIC specified the proper packtype (fieldlength) will be selected for the FDF (see above)

```
        02  &&PGRP  PIC 9(3)      COMP   * PRODUCT GROUP
        02  &&PCGS  PIC 9(7,2)    COMP   * COST OF GOODS SOLD
```

4.7. Date fields

Whenever a numeric field of 6 digits with DATE occurring in the remark is met (independent of case) this will be loaded as a datefield.

02 &&LAD   PIC 9(6) COMP   * LAST AMENDMENT DATE
02 &&EXPD   PIC DATE   * Expiry date

The first defines a field named 'Last amendment date' will be defined as a ,6, date field.

4.7.1. Year 2000

Date fields are assumed to be in the format YYMMDD as 980901 for 1998.09.01. Year 2000 dates are stored as 1YYMMDD as 1000331 for 2000.03.31.

4.8. Field occurences

The COBOL OCCURS clause defines a table of fields:

02 &&FPPB OCCURS 4 PIC S9(7,2) COMP   * PAST PERIOD BALANCE O.

A field named 'Past period balance o' will be defined 4 times in the FDF. The table facility in the FDF as a ,7,2(003) is not used as an ODBC/SQL application is better of when all fields in the table is explicitly named and can be selected directly instead of using a the table syntax:

```
SELECT AA.'Account number',
       AA.'Past period balance o',
       AA.'Past period balance o'(1) FROM AA AA
.
```

4.9. Remarks as help text

Any comment lines following a field definition is placed as field help text.

* THE CURRENT PERIOD BALANCE IS A TOTAL OF ALL

4.10. Copying definitions

A .BOOK section may include other .BOOK definitions as for

COPY DX SUBSTITUTING "DD"

The file definition for DX is copied, recordtype DX will be substituted by DD.

4.11. Redefinitions

The COBOL language often uses the redefines clause as:

01 &&CONF REDEFINES &&CO

Such redefinition causes Bxxx byte number of the field format to be set which means both the orginal definition and any following redefinitions will be loaded as fields in the FDF

4.12. Example of Date, Occurs and Redefines loading

The file AH may be defined in the copybook as:

```
.BOOK AH                 * File CAB SL header record
********************
* HEADER RECORD IN CAB FILE   - RECORD LENGTH = 352
********************
        02  &&TYP      PIC X(2)        * RECORD TYPE (AH)
        02  FILLER     PIC X(2)
*
* RECORD KEY
*
        02  &&ACC      PIC X(7)        * ACCOUNT NUMBER (ZEROES)
*
* FILE VERSION NUMBER
*
        02  &&VER      PIC 9(2,2) COMP
*
* PERIOD END DATES
*
        02  &&PDTS
           03  &&CPD  PIC 9(6) COMP  * CURRENT PERIOD DATE
           03  &&PPDS                 * PAST PERIOD DATES
            05  &&P1D PIC 9(6) COMP  * PAST PERIOD 1 DATE
            05  &&P2D PIC 9(6) COMP  * PAST PERIOD 2 DATE
            05  &&P3D PIC 9(6) COMP  * PAST PERIOD 3 DATE
            05  &&P4D PIC 9(6) COMP  * PAST PERIOD 4 DATE
*
        02 &&CPUD      PIC 9 COMP     * 0=no updates in close period
                                      * 1=CAB/SLT files updated in
                                      * close period
        02  FILLER     PIC X(323)     * RECORD LENGTH = 352
*
   01  FILLER REDEFINES &&PPDS
        02  &&PPD OCCURS 4     PIC 9(6) COMP
.END
:
```

and will load as:

No	Name	Format	ByteLayout	
1	Record type (ah)	2	0	xx
2	Filler	2	2	xx
3	Account number (zeroes)	7	4	xxxxxxx
4	Ver	2,2	11	99,99
5	Current period date	,6,	13	99.99.99
6	Past period 1 date	,6,	16	99.99.99
7	Past period 2 date	,6,	19	99.99.99
8	Past period 3 date	,6,	22	99.99.99
9	Past period 4 date	,6,	25	99.99.99
10	Cpud	1,	28	9
11	Record length = 352	323	29	xxxxxxxxxxxxxxxxx
12	Ppd	6,B16	16	999999
13	Ppd 1	6,	19	999999
14	Ppd 2	6,	22	999999
15	Ppd 3	6,	25	999999

Fields=15, Record length=352

11. Example of Date, Occurs and Redefines loading

Figure list

Index