SPEL system

Technical documentation (Rev. 1)
Vol. 1: Basics

A. Zintl
H.-J. Greuel

January 1995

The concept of the SPEL-system was developed at the Institut für Agrarpolitik, Marktforschung und Wirtschaftssoziologie of the University of Bonn by W. Henrichsmeyer, W. Wolf and H.-J. Greuel.

Theme
Agriculture, forestry and fisheries
Series
Methods

5
E

Cataloguing data can be found at the end of this publication

Luxembourg: Office for Official Publications of the European Communities, 1995

ISBN 92-826-9770-3
ISBN 92-826-9769-X (Volumes 1 + 2)

Printed in Belgium

Gedruckt auf chlorfrei gebleichtem Papier
Printed on non-chlorine bleached paper
Imprimé sur papier blanchi sans chlore

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1. Overview of the documentation

The SPEL approach is characterized by mutual interaction between model-builders and statisticians or policy-makers. There are different models to deal with different questions, e.g. the SPEL/EU-Model for the agricultural sector of the EU Member States and the EU as a whole.

In general, the SPEL System offers a systematically structured data system to be used for

- checking the consistency of agricultural statistics,
- monitoring the present situation in the agricultural sector,
- ex-post analyses of sectoral developments, and
- forecasts and policy simulations of the effects of alternative policies from short-term and medium-term viewpoints.

The set of statistical data is not taken as final but subjected to consistency checks and critical investigations, which may lead to the jointly agreed revision of existing statistics and proposals for amendments and conceptual changes for new statistics. It is not assumed, either, that policy-makers are able to specify target variables before taking a closer look at the problems involved. Instead, the fundamental idea is that target priorities emerge during the process of policy dialogue on trade-offs between target variables.

The principal technical approach for the whole SPEL System is transparency and a flexible user interface. Transparency means that each data element can be traced back to the basic data sources and the underlying assumptions. The flexible user interface facilitates dialogue between policy-makers and the model.

For the methodological aspects see 'SPEL System, Methodological Documentation (Rev. 1), Vol. 1 and Vol. 2'.

The technical documentation of the SPEL System consists of the following parts:

Basic information

SPEL System, Technical Documentation (Rev. 1), Vol. 1: Basics

This volume includes the documentation of all technical basics, as there are data dimensions, file format conventions, utilities and multi-purpose programs without any model specific code. This general software was developed by EuroCARE in co-operation with the Institut für Agrarpolitik, University of Bonn, and is applied to other uses, too.

Specific information for the models of the SPEL System

SPEL System, Technical Documentation (Rev. 1), Vol. 2: BS, SFSS, MFSS

This volume describes the generation of the SPEL/EU components Base System (BS), Short-term Forecast and Simulation System (SFSS) and Medium-term Forecast and Simulation System (MFSS) in terms of underlying methodological ideas and technical basics.

The technical documentation is designed as a guide for operators and users familiar with the VM/CMS operating system on an IBM or compatible mainframe.

1.2. Syntax conventions

When describing parameters or options, the following syntax conventions are used:

- When there is a choice between two or more parameter values, the values will be separated by a vertical bar ('|'). For example, the parameter that allows the amount of listed information for a SPEL program to be set to 'FULL', 'ROUGH' and 'NONE' will appear as follows:

  FULL | ROUGH | NONE

- Optional parameter definitions are given in brackets, and operands for which an appropriate value must be substituted are given in italics. For example, if a user wants to define a selection of table columns and rows and optionally the amount of listed information, this will appear as follows:

  COLUMNS = *columns*, ROWS = *rows*, [LSTLEVEL = FULL | ROUGH | NONE]

- However care must be taken not to actually enter the brackets and vertical bars that appear in descriptions of parameters, since they are not part of the actual parameter definition.

In the screen dumps describing panel user interfaces the following conventions are used:

- All user inputs are bold-printed; in this document they are only formal examples.

- When there is a choice between two or more cursor selections, the selectable items are also bold-printed.

2. IMPLEMENTATION

The SPEL software is designed to be implemented on the VM/CMS operating system on the Amdahl mainframe at Eurostat in Luxembourg. The programming language is FORTRAN 77. Interfaces are designed within the mainframe software to prepare exploitation data for PC standard software.

2.1. Users and operators

To become familiar with the SPEL System it is important to know about the SPEL System structure and the work levels for SPEL operators and users.

Working with the SPEL System can be done on two levels (See figure 'User and Operator level'):

- Operator level
- User level.

Operators are responsible for preparing original data and doing specific model calculations in line with the methodology.

Ex-post data and model results are offered to the users for reports and exploitations. A set of user-friendly programs is made available to users, who do not need to have detailed computer experience.

All development and revisions of SPEL software will be done by programmers.

2.2. System software requirements

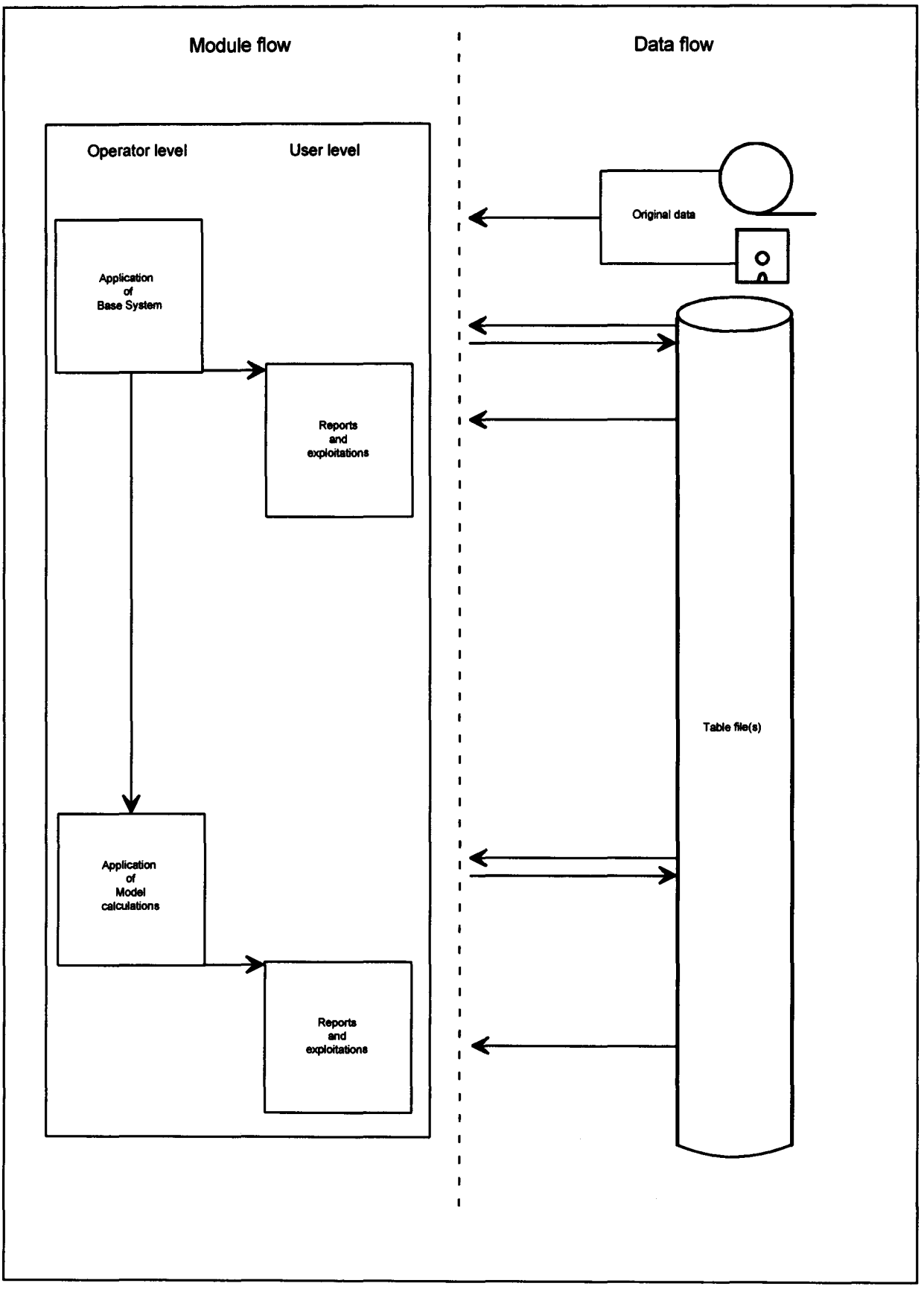
System software required:	IBM VS FORTRAN	Compiler
	IBM GDDM	Graphical data display manager
	VMSORT	Sort program
	IBM REXX	Restructured Extended Executor Language System Product Interpreter
	IBM XEDIT	System Product Editor
	MINOS	Large-scale optimization system (linear and non-linear)

2.3. System initialization and environment

To set up the SPEL software and SPEL-Data for one of the SPEL models, a special procedure must be called before using any SPEL program. This procedure is implementation-dependent and is installed and maintained in Luxembourg by the Luxembourg team. For users the initialization procedure starts a program selection shell as a user-friendly guide through the work steps.

For further detailed information see the methodological documentation of the SPEL System.

Figure 1: User and operator level



3. FORMAL STRUCTURING AND CODES

In line with methodological concepts and data flow analysis the data in the SPEL System (SPEL-Data) are logically allocated to tables, each identified by a unique alphanumeric key. This key is divided into subkeys, each directly related to six of the SPEL-Data dimensions (See figure 'SPEL-Data dimensions'). A table is defined by two dimensions, the column and row. In total, SPEL-Data are characterized by eight dimensions.

Each table element is thus defined by its column and row.

The number of rows and columns, their sequence and definition are called the table structure.

Within a model area (See below) there are two table structures: basic table structure and complementary table structure.[1]

- The basic table structure represents the modeling ideas. It is used for any results.
- The complementary table structure is more differentiated, responding to the original data and used to constitute the basic table structure.

3.1. SPEL-Data dimensions and SPEL table subkeys

Each of the eight dimensions is assigned a methodological differentiation criterion.

The size of each dimension depends on the SPEL model; for example, the number of countries in the SPEL/EU-Model is different to the number of countries in the SPEL/EU-ROW Model. SPEL-Data dimensions and SPEL table subkeys correspondence is described in the following figure.

1 See 'Methodological Documentation, Vol. 1: Basics, BS, SFSS, Part 1 and Part 2, Annexes'.

Figure 2: SPEL-Data dimensions

	Dimension	Translation rules	Table subkey	
1	Region		Region	1
2	Sub-region		Sub-region	2
3	Year		Year	3
4	Periodicity		Periodicity	4
5	Status	Base year Origin and treatment Table structure type	Base year	5
6			Type	6
6	Model area		Model area	7
7	Table column			
8	Table row			

The last two dimensions, table columns and table rows, are the dimensions of the SPEL tables.

The sequence and definition of the table rows and columns depend on the table structure type within a model area (See below).

N.B.:
In future technical revisions the dimension 'Status' will be represented by a single table subkey 'Status'.

3.1.1. Region

Regions are defined by alphanumeric codes, each of up to three characters.

For the SPEL/EU-Model the official Union codes are used for the Member States.

For example:

D	Germany
IRL	Ireland
F	France
E12	EUR 12

For the SPEL/EU-ROW Model the FAO-country numbers are used.

For example:

231	USA
059	Egypt

3.1.2. Sub-region

Two characters are used for the code. Currently the whole region is always used.

For example:

00	the whole region

3.1.3. Year

In the SPEL System, current years are used for ex-post data and projection years are used for simulation results.

The last two digits of the year numbers are used for the code.

Years 1941 to 1999 are coded as 41 to 99. Years 2000 to 2040 are coded as 00 to 40.

For example:

85	year 1985
01	year 2001

3.1.4. Periodicity

Two characters are used for the code.

00 whole year

01 ... 12	January ... December
Q1 ... Q4	First quarter ... fourth quarter
H1, H2	First half of a year, second half of a year

Currently the whole year is always used.

3.1.5. Status

The status is defined by six-character codes. The first two characters stand for the last two digits of the base year of simulation results. For the ex-post period, the code 'NN' (no base year) is always used. The next three characters specify origin and treatment and the last character identifies the table structure type for a specific model area: B=basic table structure, C=complementary table structure.

For example:

NNZPAC	Ex-post data from ZPA1 domain and belonging to table structure C
93SFSB	Result of short-term forecast(SFSS) with base year 1993 and belonging to table structure B

3.1.6. Model area

Data which are specific to one model area can be distinguished by this sub dimension. One character is used for the code.

S	Supply (SPEL/EU)
D	Demand (SPEL/EU)
E	Combination of supply and demand (SPEL/EU)
T	Trade (SPEL/EU-ROW)

3.1.7. Table column

Table columns are defined by alphanumeric codes, each of four characters.

For example:

SWHE	Production activity soft wheat
PRIC	Producer/purchase prices (unit value)

3.1.8. Table row

Table rows are defined by alphanumeric codes, each of four characters.

For example:

BARL	Product barley
LEVL	Production activity level

3.2. COD files

All lists of codes (except the codes for years, which are self-evident) are included in a COD file.

This file is used for different purposes:

- Documentation of the used codes and their sequence within the SPEL-Data dimensions. For the user this file is just a further item of information.
- Generation of program declarations supporting array indexing in the SPEL programs by codes.

The complete listings of the COD files and the methodological background for the formal structuring and coding is described in the methodological documentation for each SPEL model.

4. DATA MANAGEMENT

For data management, the SPEL System distinguishes between non SPEL data file formats and SPEL standard data file formats identified by standard file name extensions. All data which are to be imported into the SPEL System have to be converted from their own original format to a SPEL standard format.

4.1. External data sources importation

Most of the external data used in SPEL are originally selected from different data sources and each originator has its own data format.

The formats are :

Originator	Format
CRONOS	LRECL 22 (Code 18 bytes Data 4 bytes, binary floating)
FAO	FAO STANDARD FORMAT (See AGROSTAT user tape description)

4.2. SPEL standard data file formats

Within the SPEL System only a few file formats are used and all external file formats are converted. In the VM/CMS operating system the following file name extensions stand for the standard formats:

IMP and PIM
SDA
TAB
DES

Files of those formats are further called IMP files, PIM files, SDA files, TAB files and DES files.

External data on magnetic media coming from external originators is first converted to IMP format or packed PIM format, whilst data coming from listings, printed publications, experts' findings, etc. are directly edited into SDA files. ALL SPEL-Data are finally stored in a TAB file , except those stored in SDA files and being used for special program control (e.g. scenario parameters). The DES files are special applications of the TAB format used to store description text for all codes within the SPEL-Data dimensions.

4.2.1. IMP and PIM files

The IMP file format is the standard format for importing external data into the system. It is optimized for fast item selection. IMP files are the result of the format conversion program DATCON. Optionally the amount of data can be reduced by the IMPSEL program.

Data from IMP files can be directly imported into the model work file(TAB file) by the ORIGIN import program.

File structure :

IMP files are sequential files of fixed record length with a specific record structure which is unique for all records:

rrrssyymmddddccccccccfvvvvvvvvvvvvvv

where:

pos. 01-03	*rrr*	Region code
pos. 04-05	*ss*	Sub-region code
pos. 06-07	*yy*	Current year code
pos. 08-09	*mm*	Periodicity code
pos. 10-13	*dddd*	Source type code (original domain name)
pos. 14-22	*ccccccccc*	Remaining data code (up to nine characters, left aligned)
pos. 23	*f*	Data status flag as specified by originator if existent, otherwise blank
pos. 24-37	*vvvvvvvvvvvvvv*	Numerical data anywhere in the field [2] (e.g. 1.23456 -1234.5 12.3 E-10 123.E03 2E10 means $2*10^{10}$)

Example of IMP file:

```
----+----1----+----2----+----3----+----4--
D  008000ZPA10002345   185.
F  008000ZPA10002345   180.
  ...
  ...
NL 008000ZPA10002345   180.4
----+----1----+----2----+----3----+----4--
```

The records are sorted according to columns 01-22 in character ascending order.

The remaining data code, the data status flag and source type code [3] are usually the same as the code used in original data sources.

IMP files are normally very large files. In order to spare disk space those files can be converted by the IMPPACK program into a packed format of type PIM and vice versa. They cannot directly be managed by editors or listed using view/browse commands. PIM files can be input for the IMPSEL program.

---

2 FORTRAN F14.0 format

3 In FAO data, no originator type code is included. The code is set by a parameter of format conversion program (DATCON).

4.2.2. SDA files

The SDA file is used for all external data to be edited by the operator/user.

The data representation is in the form of tabulated sequential time series. SPEL table subkey codes (e.g. region, sub-region etc.) and SPEL table element codes (for column and for row) are supported. All subkey codes and table element codes are unequivocally interpreted. Additional comment lines are very helpful for the reader.

File structure:

SDA files are sequential files of fixed record length and record structures which differ for six record types:

Comment records

begin with an '*' at the first character position and may occur anywhere in the file.

The file header record

is used to identify the file as an SDA file. It contains

```
$STANDARD
```

at the beginning of the line and must be the first non-comment record in the file.

Table header records

are used to mark the beginning of new time series tables.

```
$TABLE   rrrss....bbddddm
```

where:

pos. 01-06	`$TABLE`	Fixed identifier
pos. 07-10		*Blank*
pos. 11-13	*rrr*	Region code
pos. 14-15	*ss*	Sub-region code
pos. 16-19	`'....'`	Fixed[4]
pos. 20-21	*bb*	Base year code, currently 'NN' for ex-post data
pos. 22-25	*dddd*	Type code
pos. 26-26	*m*	Model area code (S=supply, T=trade, D=demand, E=supply&demand, (*) =undefined)

4 Filler to have the same layout as SPEL table keys (here years and periodicities)

Column definition records

define the codes for the data columns in the following data records. They must follow the table header record and be written before data records. Additional column definition records may occur whenever data line column codes change.

`$COLUMNS  ccccllll [ccccllll ccccllll ccccllll ....]`

where :

pos. 01-08	`$COLUMNS`	Fixed identifier
pos. 09-10		*Blank*
pos. 11-18	*ccccllll*	Code for data column 1 `cccc` - model table column code `llll` - model table row code
pos. 19		*Blank*
pos. 20-27	*ccccllll*	Code for data field 2
...		
pos. 65-72	*ccccllll*	Code for data field 7

Data records

are used for enter numerical data.

Structure:

yymm *aaaaaaaaa*[*bbbbbbbbbccccccccddddddddd...*]

where :

pos. 01		*Must be blank*
pos. 02-03	*yy*	Current year code
pos. 04-05	*mm*	Current periodicity code
pos. 06-09		*Blank*
pos. 10-18	*aaaaaaaaa*	Numerical data field 1. The data may be anywhere in the field and must include a decimal point (e.g. 1.23456 ; -1234.5; 12.3 E-10; 123.E03; 2.E10 means $2*10^{10}$) If the field is totally blank, it is interpreted as non-existing in the file
...		
pos. 64-72	*ggggggggg*	Numerical data field 7.

The end record

must be

`$END`

at the beginning of the line and must be the last non-comment record in the file.

Example of SDA file[5]:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
$STANDARD
*
*     Process level for Germany
*
$TABLE    D  00....00ZPACS
$COLUMNS  LEVLSWHE LEVLDWHE LEVLBARL
 8600         99.9     99.9     99.9
 8700         99.9     99.9     99.9
*
*     Process level for France
*
$TABLE    F  00....00ZPACS
$COLUMNS  LEVLSWHE LEVLDWHE LEVLBARL LEVLOATS LEVLMAIZ LEVLOCER LEVLPARI
*8400
 8500         99.9     99.9     99.9     99.9     99.9     99.9     99.9
 8600         99.9     99.9     99.9     99.9     99.9     99.9     99.9
 8700         99.9     99.9     99.9     99.9     99.9     99.9     99.9
$COLUMNS  LEVLPULS LEVLSUGB LEVLRAPE
 8400         99.9     99.9
 8500         99.9     99.9
 8600         99.9     99.9     99.9
 8700         99.9     99.9     99.9
$END
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
```

4.2.3. TAB files

The TAB files are work files and contain original data as well as results of all SPEL work steps as data description text. Storing of all intermediate results permits re-execution at any time from any point in the work step sequence.

TAB files are logically divided into entries. Each entry is divided into three parts. First the entry key, second an entry text area and third an entry data area.

For example in SPEL an entry key 'UK 00800000BASB' means that data is stored for: region UK, sub-region 00 (the whole region), current year 80 (1980), periodicity 00 (the whole year), base year NN (no base year), type BASB (result of work step BASEMOD, basic table structure), model area S (supply).

General maintenance for TAB files can be done by the menu-driven utility program DASERV. It offers services for creating new files, listing the keys of all entries, printing entries, reorganizing files, copying and renaming entries as well as many others.

All access of programs to these TAB files is done by the TAB file manager, which forms part of the general programs.

A SPEL user never has to consider the format or internal file structure but has to consider the following:

- A file is static in its size, which is definitively fixed when created. Estimates of the maximum number of entries and the size of entries to be stored must be made when the file is created.

---

5 The values in the table are for illustration only.

- Before accessing a new file by programs, the key structure should be defined or copied from another TAB file by using the appropriate DASERV service.

- A new entry is implicitly created during the first writing operation using the corresponding table key. The file size occupied by the entry is determined by the key size, an optional comment field size and the number of non-zero table elements. Zero elements in 'sparse' tables do not occupy any disk space. The file size occupied by an entry is always continuous.

- During subsequent writing with the same entry key, the same location will be used as long as the new entry size fits the old one. If the new size exceeds the space reserved for the already existing entry, the latter is implicitly deleted and a new entry created.

- File areas occupied by entries implicitly or explicitly deleted by user action cannot be used immediately for new entries. This 'unused space' must be freed by running a service of the DASERV utility.

- Utilization factors for numbers of entries in the file directory and for the stored data records are printed at the end of the LOG file of application programs; they can also be listed by the DASERV utility.

- To enlarge an existing table file, a new table file of sufficient capacity must be created first. The entries in the old table file can then be copied into the new file by using the DASERV utility.

4.2.3.1. TAB files for SPEL tables

New TAB files for SPEL tables, called table files, must be created by DASERV because SPEL programs can only access already existing TAB files.

Using files of TAB format, each entry is a 'SPEL table' as described in the chapter 'Formal structuring and codes'. The file structure ensures an acceptable execution speed because tables are defined to match the data flow of SPEL programs. The SPEL tables include those data mostly accessed simultaneously. This procedure therefore allows minimum disk access activities and high flexibility.

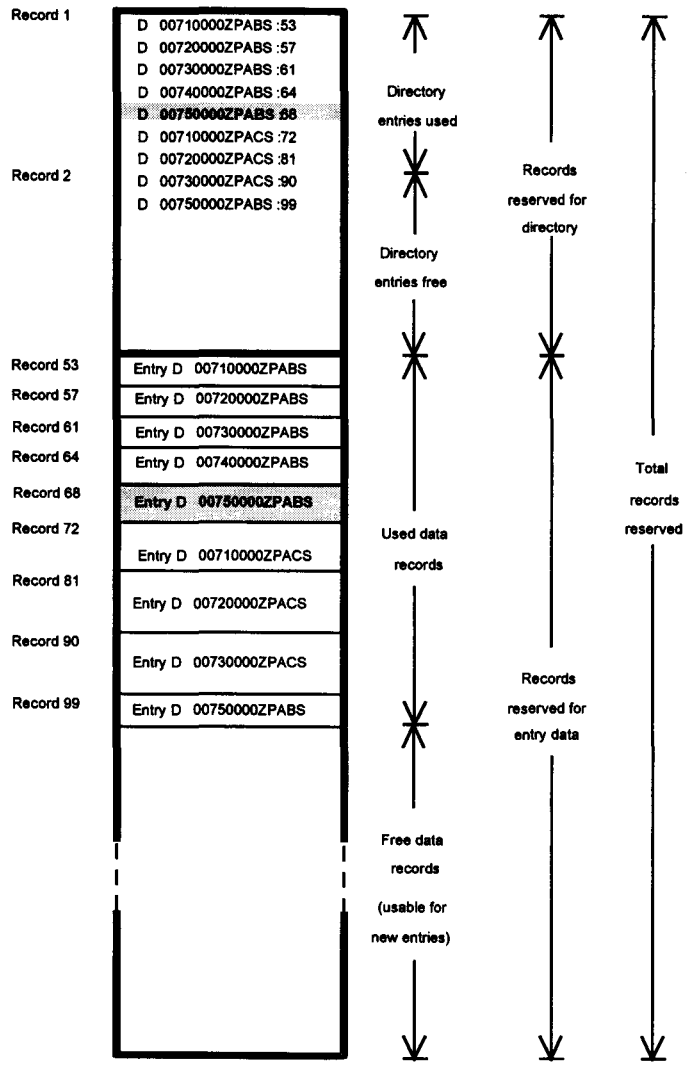
The entry key of each SPEL table consists of the codes identifying the SPEL-Data dimensions, e.g. 'E12 00940093SFSBD' means: region E12, sub-region 00 (the whole region), current year 94 (1994), periodicity 00 (the whole year), base year 93 (1993), type SFSB (result of short-term forecast, basic table structure), model area D (demand).

The entry text area, here only comment, includes the name of the program and date and time of the last write access.

The entry data area includes the numerical data of the SPEL tables.

In the VM/CMS operating system these files have the name extension TAB.

Figure 3: Structure of TAB files



Notes:

An entry may consist of text or numerical data or both. The numerical data part is an array of data elements. In SPEL it is a table of size rows by columns; the text part is comment (e.g. including creation data and time). The table is interpreted as a vector of length rows multiplied by columns. To minimize storage space, the vector is packed (e.g. zero data are removed) and stored as a flow without any regard for record boundaries.

The entry references in the directory formed from the entry key and the number of the first entry record) are also stored as a flow.

In this figure "deleted" entries are marked by shading.

4.2.3.2. DES files for description text

Description text for all codes used within the SPEL-Data dimensions are also stored in files of TAB format, called DES files. These files are generated by the DESCRI program (See chapter 'General programs').

Each description entry is defined by a unique key. In SPEL, an entry key 'SBSWHELEVL' means: model area S, table structure type B, table element defined by the combination of column and row code SWHELEVL.

The entry text area includes the description text, e.g. for the entry key 'SBBULPGVAM' the text 'Male adult cattle, GVA at market prices'.

The entry data area is empty.

In the VM/CMS operating system these files have the file name extension DES.

5. USER INTERFACE

The user interface comprises panels, general control files and special control files.

Special control files permit large sets of control statements which cannot be modified during run-time.

5.1. General program control

General program control is done by parameters. Each parameter has a unique name and its current value is defined by an assignment.

parameterName = parameterValue

The *parameterValue* is interpreted according to the application. The specific usage of the parameters is described in the documentation of each SPEL program.

Most SPEL programs require the selection of SPEL data dimensions by codes. For these selections, the user often has to enter more than one code as *parameterValue,* e.g. if the user wants to select codes for several regions. Depending on the application program, SPEL offers the following general options:

- a sequence of codes, such as 'D E F GR'
- an alphanumerical range defined by a lower and upper bound, such as 'A : Z'
- a logical range defined by a lower and upper bound, such as 'SWHE - OCER', according to the methodological code sequence as defined in the COD file
- a combination of ranges and sequences, such as 'SWHE - BARL MAIZ OCER'
- a star ('*') to select all items

Presets for all these parameter values are loaded from general control files. The parameters can be modified during run-time by panels.

5.1.1. General control files (PAR files)

General control files have the name extension PAR, further called PAR files. Once the SPEL System is implemented, a DEFAULT PAR file is installed by the operator. For all SPEL programs, default presets for the program control parameters are included. The operator and user can create their own version with adapted parameters for special program applications.

PAR files are sequential files of a fixed record length of 80 characters in free format. The control parameters are grouped into parameter blocks according to their usage.

There are four different record structures:

- Comment records

Beginning with an '*' at the first character position and may occur anywhere in the file.

- The parameter block header record

This record includes the name of the parameter block. It is a character string of up to 20 characters in length. It indicates the beginning of a parameter block.

- Parameter definition record(s)

Each record includes one assignment of a parameter value to a parameter name.

name = value

where *name* and *value* are character strings. The whole assignment must not exceed 80 characters. The parameters belong to a block defined in the previous block header record.

- Parameter block end record

The end line must be

```
END
```

Example of part of a PAR file:

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
ORIGIN
 Program text =Import original data
 LOG file (LOG) = ORIGIN LOG A
 Local help file (HLP) = ORIGIN HLP *
 Output file (TAB) = SPEL-USR TAB A
 Model area = S
 Assign file (ASS) = * ASS *
 Input file (SDA/IMP): 1 = * SDA *
 Input file (SDA/IMP): 2 = * SDA *
 Region = D  F
 Current year = 72 : 90
 Type = ZPAC EXPC COSC PRAC
 Additional parameter blocks = SPEL
 Additional parameter blocks = ORIGIN  GENERAL PAR I
END
#SPEL
 System text =SPEL
 Subkeys 1='Region' 'Sub-region' 'Current year' 'Periodicity' 'Base year'
 Subkeys 2='Type' 'Model area'
 Subkey length= 3 2 2 2 2 4 1
 Model areas =D S E T
 Area text D =Database for model area DEMAND
 Area text S =Database for model area SUPPLY
 Area text T =Database for model area TRADE
 Area text E =Database for model area SUPPLY&DEMAND
END
#DEFAULT
 LOG file (LOG) =SPEL LOG C
 Global help file (HLP) =SPEL HLP *
 Description file (DES) =SPEL DES F
END
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
```

Special parameter 'Additional parameter block'

Every main program has its own parameter block, the name of which is similar to the program name. General parameters, such as LOG file names, are collected in separate parameter blocks. Access to these blocks can be provided by the special parameter 'Additional parameter block' defining an additional parameter block to be loaded.

For this special parameter the parameter value is the name of the additional block to be loaded. If the additional parameter block is part of a different PAR file, the file name has to be appended to the block name. In the example above, the block SPEL is to be loaded from the same file and the block ORIGIN is to be loaded from the file 'GENERAL PAR I'.

One or more 'Additional parameter block' definition records may occur immediately before the end record (See example). The parameters appearing in the parameter block corresponding to a main program will always have priority if the parameter is also defined in an additional parameter block.

5.1.2. Panels

Panels are screens allowing user interaction or showing messages.

Three kinds of panels can be distinguished:

- edit panels (control parameter and data edit)
- selection panels (item selection with cursor selection facilities)
- display panels (messages, help and data display).

When calling a program, the user will get a sequence of screen panels to verify and optionally update all program control parameters. Presets for input fields and top line text, except control level text, come from PAR files. The panel layout is standardized.

The text which appears in bold in the panels can be either overtyped by the user or selected by cursor for the current run. The following examples show typical panel layout.

Edit Panel

The top line of the panel shows on the left the purpose of the running program, in the middle the current software name and possibly the version, and on the right the control level. The control level text issued by the program may be replaced by error messages and warnings. The active function keys are displayed at the bottom.

The center of an edit panel contains parameter fields with the parameter name on the left and the parameter value on the right. The bold-printed parameter values can be updated by user input. By pressing the ENTER key the input is tested for acceptability. When errors are detected, a short error message will appear at the top right and the cursor will be placed on the erroneous input for correction.

Example: Work file selection

```
Import original data -------------------- SPEL ----------------- Work file selection

                              Please enter file names




   Output file (TAB)       => SPEL-USR TAB A
   Assign file (ASS)       => * ASS *
   Input file (SDA/IMP):1 => * SDA *
   Input file (SDA/IMP):2 => * IMP *
   Input file (SDA/IMP):3 =>
   Input file (SDA/IMP):4 =>
   Input file (SDA/IMP):5 =>
   Input file (SDA/IMP):6 =>
   Input file (SDA/IMP):7 =>
   Input file (SDA/IMP):8 =>




                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

The example panel is used by the program ORIGIN to control all files used. The file names are tested for conformity and processing mode.

Between the top line and the center additional header lines and between the center and the function key description lines, additional bottom lines may appear as further comments (e.g. 'Select work file'). In any file selection panel, file names may be specified using wild card characters such as '*'. If a file name contains wild cards, a list of matching files is displayed in a special screen (panel). Selection is effected by placing the cursor on the desired file name line and pressing ENTER.

Selection Panel

The top and bottom lines have the same structure and the same functions as in the edit panel. In the center of a selection panel there is a list of character items; below, a list of ASS files is shown. To select one of these items the user has to place the cursor on the item line and press ENTER.

Example: Work file selection

```
Import original data ------------------ SPEL -------------------- Work file selection

                              Assign file name selection




                                   SUPOLD   ASS   J
                                   SUPPLY   ASS   J
                                   DEMAND   ASS   J
                                   DEMOLD   ASS   J
                                   PORTUGAL ASS   J




                  Select a file by moving the cursor to any file name above

                            Enter= ok  1= Help  3= Quit  4= Exit
```

Display Panel

Display panels only show text. They are also used to display both messages when running a SPEL program and help texts (See chapter 'On-line help information').

Example: Display of a LOG file

```
 * Working for Member State: D   sub-region: 00
-
 **** Get regression steering file
-
 *** 240  records selected from regression steering file


 **** Get time series
-
 *       Working for type: ZPAC
-
 *       Working for type: COSC
-
 *       Working for type: PRAC
-
 *       Working for type: EXPC
-
 *       Working for type: ESTC
```

The example above depicts the log panel when running the ORIGIN program.

List of default function keys :

Function key no.	Use of General function keys	Use of Paging facilities function keys
1	Help (call Help panel)	-
3	Quit (back to previous control level)	-
4	Exit (program exit)	Find (search for keyword)
5	-	Top (top of file)
6	-	Bottom (bottom of file)
7	-	Backward (one page back)
8	-	Forward (one page forward)
9	-	Left (go to left side)
10	-	Right (go to right side)
11	Save/Load (save or load parameters)	-
12	General files (modify name of general file)	-

The bottom lines of each panel show the available function keys. For some applications the meaning of function keys may change (e.g. function key 4). Function key 1 (Help) and function key 3 (Quit) are invariable.

5.1.3. On-line help information

Context-sensitive on-line help is available for each field of edit and selection panels. The user merely has to place the cursor on any displayed text and press the help key (function key 1).

On-line help is also available for error messages currently displayed. After pressing the help key, more detailed information is displayed in a display panel.

If, in the edit panel example given above, the user had placed the cursor on the text 'Table file' and had pressed function key 1 the following screen would be displayed.

Example : Help information screen

```
 ------------------------------------------------------------ Help information

-Table file
    This file is used for output and must be of TAB format (SPEL work file)

    All data which fit the selection criterion (next panel) will be
    stored as 'table' entries in this file

    To select a file from a list of existing files please use
    wildcard characters such as '*' (e.g. '* TAB *')

    For further detailed information about SPEL data file formats
    see 'Data Management' chapter in the Technical Documentation.









                              Enter= ok  3= Quit  4= Find
```

To search for keywords in a large help text, a 'Find' key is defined. When the information does not fit on to one screen, paging facilities are offered.

All help texts are collected in help files and are used by the on-line help system.

Two help files are accessed by each program:

- the local help file
  includes help information which is specific to the program.

- the global help file
  includes help information used by several programs.

The names for the help files are defined in the parameter file.

The user/operator neither needs to know the structure of the help files nor has to change any text in those files.

5.1.4. Parameter file link panels

As already mentioned above, presets for all control parameters used in the panels are loaded from a general control file, which is structured in named parameter blocks. The DEFAULT PAR file is used when the program is called. If the user wants to use his own PAR file, he can press the SAVE/LOAD-key to do this.

Every main program has its own parameter block, the name of which is similar to the program name. Within the parameter block, all program-specific control parameter names are in accordance with the input field description displayed on the screen. If a parameter is not found, the related panel input field preset is blank.

General parameters, used for most SPEL program panels, such as LOG file names, are collected into separate parameter blocks.

In the example in the chapter 'General control files', part of a PAR file is shown. The parameter block called ORIGIN is the one specifically used by the ORIGIN main program. The last parameters 'Additional parameter blocks' extends access to the 'SPEL' parameter block of the current file and an secondary block 'ORIGIN' in another PAR file.

If the user has changed some parameter values while revising the control panels, he can store the new parameter settings by pressing the function key 11=Save/load key. An edit panel will be displayed to define the name of the parameter file. All parameter values will be updated in the blocks they come from, with the exception of those coming from protected blocks with a name beginning with '#', like '#SPEL'. If protected parameters have changed, they will be added to the program block. In the example the parameter 'LOG file' originally in the '#DEFAULT' block will be saved in the 'ORIGIN' block, if the value has changed.

Example : Save/load panel

```
Import original data ------------------- SPEL ------------------ Save/load parameter

                            Please enter PAR file name


         SAVE    = Save the current parameters to a parameter file
         LOAD    = Load parameters from a parameter file




 Parameter file (PAR) => DEFAULT PAR F




                   1= Help  3= Quit  10= Update   11= Save  12= Load
```

If the user has decided to load another parameter scenario while revising the control panels, he can press the function key 11=Save/load key too. An edit panel will be displayed and the user can define the name of a new parameter file to be loaded. The program will start again at the first control panel.

5.2. Special control files

The SPEL programs often use a large amount of control information. It is quite clear that all these parameters should not be handled by panels. Special control files are allocated for retrieving them.

These files are sequential files including a sets of steering statements. Each steering statement has a unique syntax as follows:

keyword $parametername_1$=$parametervalue_1$ [,$parametername_2$=$parametervalue_2$, ...] ;

The substitutes for *keyword*, $parametername_i$ and $parametervalue_i$ are depending on the SPEL program to be steered, e.g. for the COMPLET program:

keyword — Steering instruction defining the mode of the steering statement, e.g. SELECT or DEF.

$Parametername_i$ — Name of a steering parameter, e.g. COLUMNS to define a selection for column codes.

$Parametervalue_i$ — Value of a steering parameter, e.g. 'SWHE - OCER' defining a selection of codes.

; — Semi-colon used as steering statement delimiter.

The following file name extensions are used:

ASS files for data selection used by the IMPSEL, ORIGIN and SFPROP programs
CST files for estimation control used by the COMPLET program
DEF files for definitions of text for SPEL codes used by the DESCRI program
PST files for the definition of exogenous variables used within the projection models, e.g. MFSS

There are still two special control file types with a dissimilar record structure, which will be adapted in future technical revisions:

STE files for the EV evaluation utility
RST files for trend estimation control used by the TREND and SFPROP programs

All kinds of special control files are described within the documentation of the program used for, e.g. the DEF control files are described in the chapter for the DESCRI program.

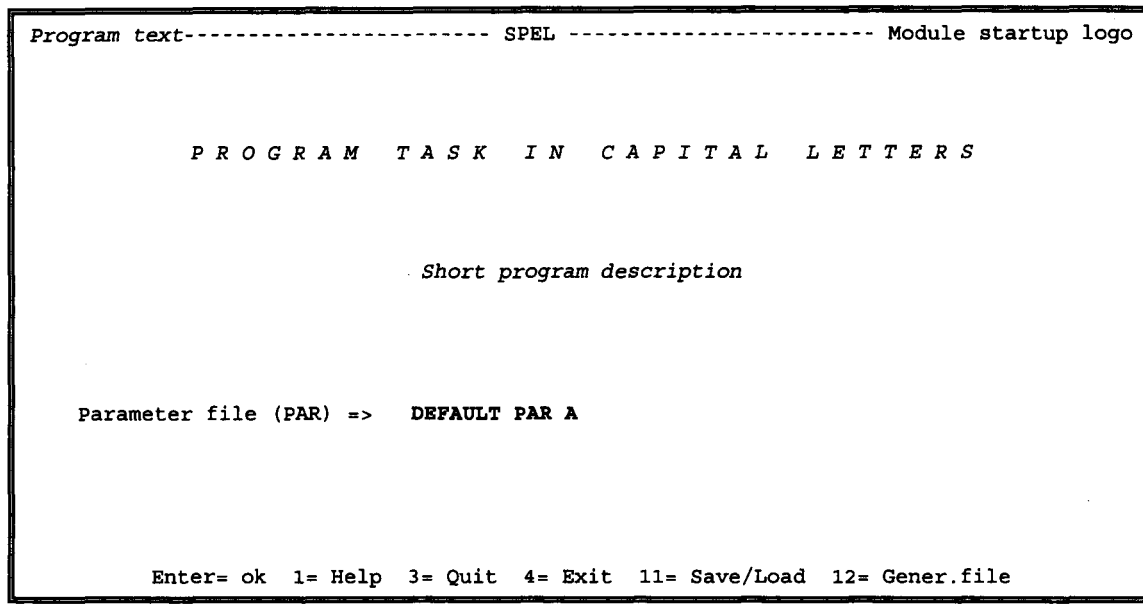
6. GENERAL PROGRAMS

This volume includes the description of general programs which are used in all SPEL models.

Their application sequence and all methodologically adjusted programs are described in the specific model documentation 'SPEL System, Technical Documentation (Rev. 1), Vol. 2: BS, SFSS, MFSS'.

Each SPEL program uses panel interfaces when starting a program, the first panel will always be the 'Module startup screen'. The layout of this panel is standardized.

Module startup screen



This screen gives a short program description and prompts the user for the parameter file name. By pressing the function key 12, an edit panel will be displayed which will allow names of general files to be changed. Depending on the application program, the file names for run time messages and warnings (LOG file), more detailed information for the operator (LST file), code descriptions (DES file) and on-line help information (local HLP file and global HLP file) can be updated.

In the LOG file the user will find a copy of all messages displayed on screen during program execution. More detailed information about work steps will be included as well. In the LST file the user will find listings as results of program tasks. The description file is a file of TAB format and includes SPEL code description text to be used for messages and listings. The HLP files are accessed when the user presses the help key.

Parameters

Parameter file (PAR)

> PAR file containing the parameter blocks, including the program parameters for the program start.

6.1. Work on TAB files

6.1.1. DASERV program: maintenance of TAB files

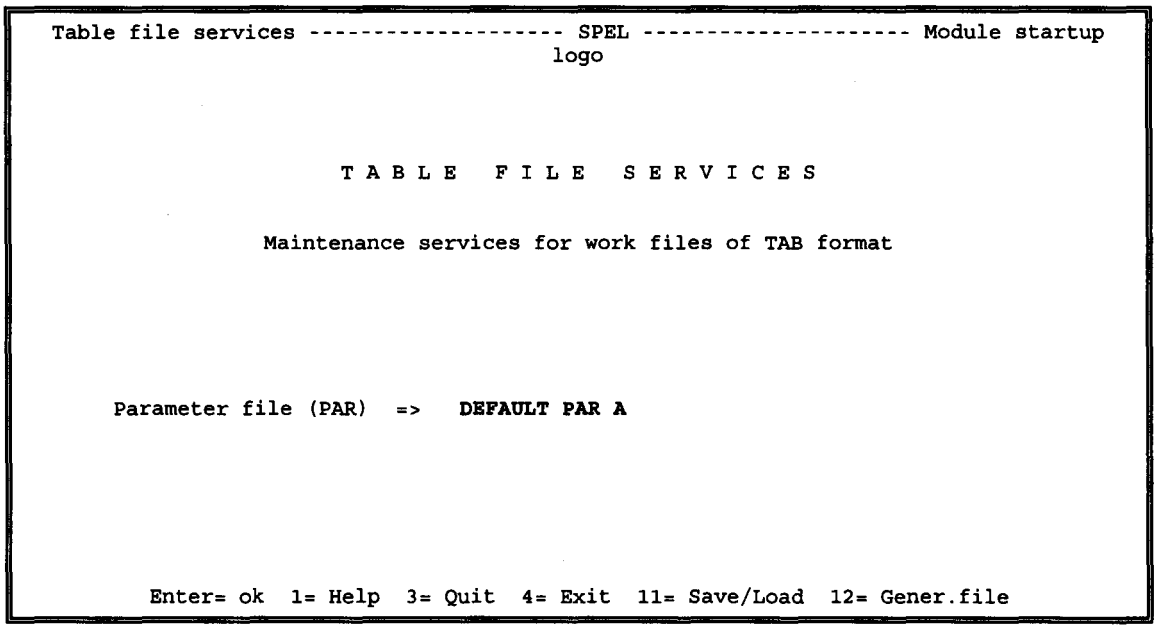
Within SPEL most numerical data are stored in files of TAB format. A special application of this TAB format is the DES file which is used to store description text for codes. The TAB format cannot be managed by editors or listed using view/browse commands.

The DASERV program offers, for TAB file maintenance, many services:

- organization and reorganization of the file
- listing and checking of the file directory (table of contents of entries)
- listing and printing of data.

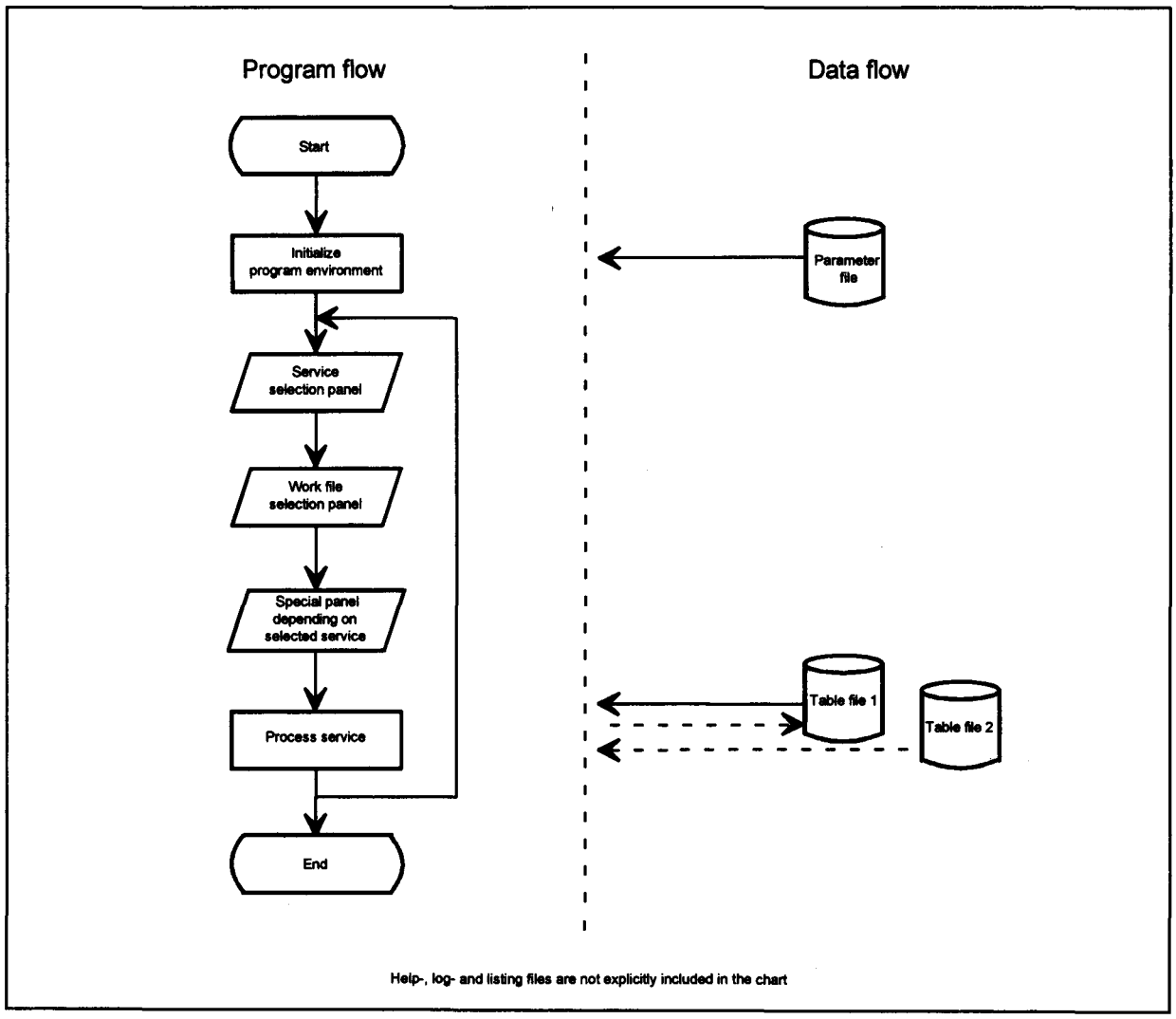
In general the DASERV program works on TAB files, but DES files can also be processed. The user has to keep in mind that some services of DASERV are not useful to work on entries including only an alphanumerical text and no data.

Module startup screen



For detailed information, see chapter 'General programs'.

Figure 4: DASERV program flow chart



Service selection screen

```
Table file services -------------------- SPEL -------------------- Service selection

                          Please select the general service

                        Authorize access
                        Compare entries
                        Copy entries
                        Copy key structure
                        Create new file
                        Define key structure
                        Delete file
                        Delete entries
                        Edit entry data
                        Free unused space
                        List entry data
                        List file directory
                        List file status
                        Print entry data
                        Rename entries
                        Test file consistency


          Select a service by moving the cursor to any of the above options
                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Selection items:

Authorize access

This service lets you define access restrictions for a specific file. You can define different groups of users for read and write access.

Compare entries

This service lets you compare the contents of entries which may be in the same or two different files.
The program outputs a list of deviations found.

Copy entries

This service lets you copy a selection of entries from one file into another or into the same file.
You may specify new subkey names for the destination entries.

Copy key structure

This service lets you copy the key structure information (name and length of subkeys) from one file into another.
The length of the whole key in the source and destination files must be equal.

Create new file

This service lets you create a new TAB file for direct access to data and text entries.

Define key structure

This service lets you define a key structure which divides the table key into up to ten subkeys.

Delete entries

This service lets you delete entries in a file. It does not free the file space occupied by the deleted entries.

Delete file

This service lets you delete an existing TAB file.

Edit entry data

This service lets you edit the data area of existing TAB entries.

Free unused space

This service lets you free unused space, i.e. file directory space and file space occupied by 'deleted' entries.

List entry data

This service lets you list the data area of TAB file entries.

List file directory

This service lets you list the entry names (keys) and entry text of a selection of entries in a file.

List file status

This service lets you list information about the file characteristics including capacity and usage information for the number of entries and number of records.

Print entry data

This service lets you print the data of TAB file entries as 'tables'.

Rename entries

This service lets you list the data area of TAB file entries. This service lets you rename entries in a TAB file. This means that the key will change without copying or moving any data.

Test file consistency

This service lets you test the file directory for inconsistencies.

Authorize access:

This service lets you define access restrictions for a specific file. You can define different groups of users for read and write access.

You will be prompted for user IDs for read and write access for the specified file. You also have to define a password to control further access authorization modifications for the specified file.

Work file selection screen

```
Table file services ---------------------- SPEL --------------------- Authorize access

                                Please enter file name




  File to be processed (TAB) => SPEL-SYS TAB F




                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

File to be processed (TAB)

The name of an existing file for which the access authorization is to be defined.

Password screen

```
Table file services ---------------------- SPEL --------------------- Authorize access

                        Please enter authorization password




                              Password =>




                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Password

The password allows changes to access authorization. It is not displayed.

User names screen

```
Table file services --------------------- SPEL ---------------------- Authorize access

                                Please define user access


Read only user  => USERID1 USERID2 USERID3 USERID4 USERID5 USERID6 USERID7
                => USERID8 USERID9
                =>
                =>
                =>
                =>
                =>
                =>
                =>
Read/write user => USERID11 USERID12
                =>
                =>
                =>
                =>
                =>
                =>
                =>
                =>
                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Read only user

List of user IDs for which read-only access is allowed.
The IDs have to be separated by blanks (e.g. `USERID1 USERID2 USERID3...`).
You may use up to 9 lines from the beginning.

Read/write user

List of user IDs for which read and write access is allowed.
The IDs have to be separated by blanks (e.g. `USERID11 USERID12`).
You may use up to 9 lines from the beginning.

New password screen

```
Table file services ---------------------- SPEL --------------------- Authorize access

            Please enter new authorization password if you want to change




              New password or blank for no change =>

              If new password, please repeat here =>




                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

New password or blank for no change

The new password for further changes to access authorization. To keep the old password, do not enter anything in this panel.

If new password, please repeat here

If a new password was entered in the previous line, it has to be repeated here to prevent misspelling.

Compare entries:

This service lets you compare the contents of entries which may be located in the same file or in two different files. The program outputs a list of discrepancies found. You will be prompted for the two file names and for an entry key selection.

Work file selection screen

```
Table file services ---------- SPEL ---------- Compare entries

                          Please enter file names




Reference file (TAB) => SPEL-USR TAB A

Compare file (TAB)   => SPEL-USR TAB C




          Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Reference file (TAB)

The name of an existing file which contains the reference entries.

Compare file (TAB)

The name of an existing file which contains the entries to be compared with the reference entries. This file may be the same as the reference file.

Entry key selection screen

```
Table file services ---------------------- SPEL ---------------------- Compare entries

                              Please enter key selection


Region        ( 3 ch.) => D F
         compare with =>
Sub-region    ( 2 ch.) => 00
         compare with =>
Current year  ( 2 ch.) => 85 : 89
         compare with =>
Periodicity   ( 2 ch.) => 00
         compare with =>
Base year     ( 2 ch.) => NN
         compare with =>
Type          ( 4 ch.) => EXPC
         compare with =>
Model area    ( 1 ch.) => S
         compare with =>


                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

The parameter names 'Region', 'Sub-region', 'Current year', 'Periodicity', 'Base year', 'Type' and 'Model area' are the subkey names defined in the service 'Define key structure'.

For each subkey selection specify a single code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

compare with

The subkeys to be compared with the subkeys specified in the previous line have to be specified in this line.
You may enter one code or a sequence of codes for each subkey referring to the compare file. When using sequences, the n'th code in the sequence specifies the compare code for the n'th reference code. The number of codes of both sequences therefore has to be equal. Ranges are not allowed. The selection line before has to contain the corresponding subkey codes referring to the reference file.
If there is no input in this line, the same subkeys as specified in the line above are used for comparison.

Copy entries:

This service lets you copy a selection of entries from one file into another or into the same file. You may specify new subkey names for the destination entries.
You will be prompted for the file names and for an entry key selection.

Work file selection screen

```
Table file services ---------------------- SPEL ------------------------ Copy entries

                                Please enter file names









Source file (TAB)       => SPEL-USR TAB A

Destination file (TAB) => SPEL-USR TAB C









                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Source file (TAB)

The name of an existing file containing entries to be copied.

Destination file (TAB)

The name of an existing file into which entries are to be copied. This file may be the same as the source file.

Entry key selection screen

```
Table file services ---------------------- SPEL ------------------------ Copy entries

                              Please enter key selection



Region        ( 3 ch.) => D F
               copy to =>
Sub-region    ( 2 ch.) => 00
               copy to =>
Current year  ( 2 ch.) => 85 : 89
               copy to =>
Periodicity   ( 2 ch.) => 00
               copy to =>
Base year     ( 2 ch.) => NN
               copy to =>
Type          ( 4 ch.) => EXPC
               copy to => XXXC
Model area    ( 1 ch.) => S
               copy to =>


                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area, see 'Compare entries: - Entry key selection screen'.

copy to

The subkeys to be copied from the subkeys specified in the previous line have to be specified in this line.
You may enter one code or a sequence of codes for each subkey referring to the source file. When using sequences the n'th code in the sequence specifies the copy code for the n'th reference code. The number of codes of both sequences has therefore to be equal. Ranges are not allowed. The selection line before has to contain the corresponding subkey codes referring to the source file.
If there is no input in this line, the same subkeys as specified in the line above are used for copy.

Copy key structure:

This service lets you copy the key structure information (name and length of subkeys) from one file to another. The length of the whole key in the source and destination files must be the same.

Work file selection screen

```
Table file services ---------------------- SPEL ------------------- Copy key structure

                                Please enter file names




Source file (TAB)      => SPEL-SYS TAB F

Destination file (TAB)=> SPEL-USR TAB A




                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Source file (TAB)

The name of an existing file containing key structure information.

Destination file (TAB)

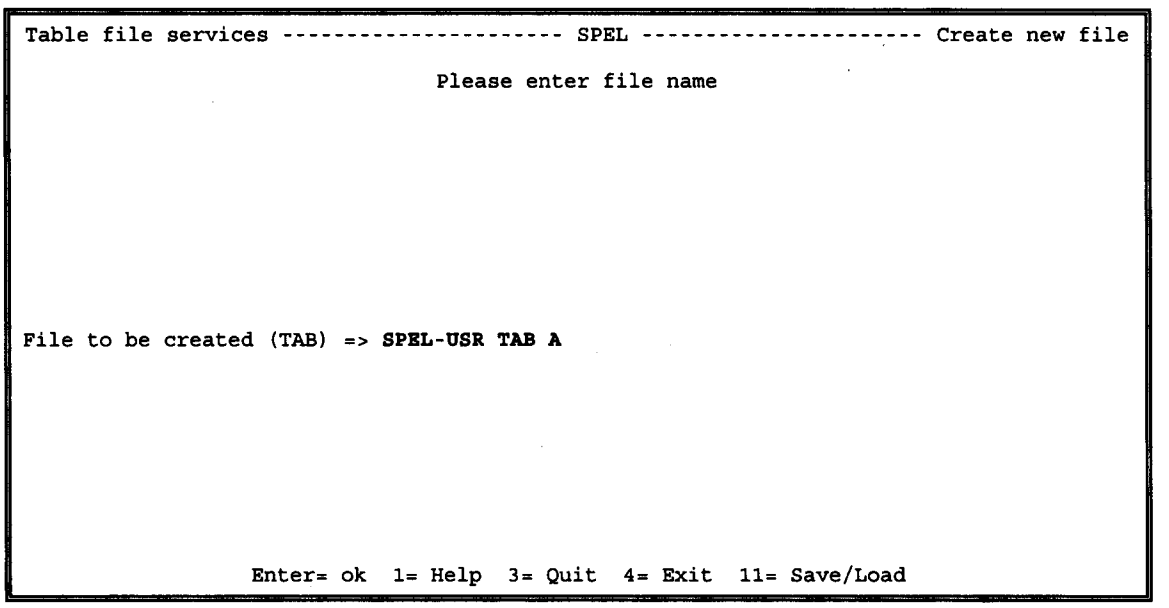
The name of an existing file into which the key structure information is to be copied.

Create new file:

This service lets you create a new TAB file for direct access to data and text entries. You will be prompted for the name of the file to be created and information about the intended use. This information to be entered is important for determining the file space to be reserved for the new TAB file. It is advisable to reserve sufficient space, but do not overestimate and waste disk space.

After the file has been created successfully, the entry key structure should either be defined by the 'Define key structure' service or copied from an existing file by the 'Copy key structure' service.

Work file selection screen



Parameters:

File to be created (TAB)

A legal file name in the current operating system for the file to be created. The file must not already exist.

File parameter screen

```
Table file services ---------- SPEL ---------- Create new file

                    P a r a m e t e r s  of  n e w  f i l e

     Length of entry key in characters                    => 16

     Maximum number of entries total                      => 5000

     Max. length of entry description text in characters  => 80

     Minimum number of data elements in an entry          => 1

     Maximum number of data elements in an entry          => 3000

     Estimated average number of data elements in an entry => 500

     Logical record length                                =>

     Number of records to be reserved                     =>

   The last two fields can not be edited. The items are computed from of the other
         fields. Pressing the function key 5 will show the current values
          Enter= ok  1= Help  3= Quit  4= Exit  5= Compute  11= Save/Load
```

Required parameters:

Length of entry key in characters

The number of characters in the entry key. The key length is fixed for all entries in the file.
According to SPEL key-structure rules, the length of the entry key has to be 16 characters. (See chapter 'Formal structuring and codes')

Maximum number of entries total

This number will be used internally for computing requirements to reserve space for data in the created TAB file, and thus determines the size of the work file.
No more entries will fit in the created file. The number given is internally rounded up to optimize disk space usage.
Example:

The new file should have space for 20 years, 15 regions and 10 types, all in all
20 * 15 * 10 = 3000 entries

Additional entry space should be reserved for 'deleted' entries and to allow storage of more tables in future.

Max. length of entry description text in characters

Maximum number of characters in the entry description text associated with each key.
The legal range is 2 to 255.
According to SPEL rules, the comment text to be stored with each SPEL table is 80 characters in length.

Minimum number of data elements in an entry

The minimum number means the lowest number of data elements over all entries to be stored in the file.
The number specified has to be lower than, or equal to, the number actually occurring.
SPEL 'basic table structure' and 'complementary table structure' tables are to be stored in TAB files. In SPEL, the storing mode is always 'packed' (zero data are not explicitly stored). The number of data elements in the entries therefore differs.
In extreme cases the number of data elements explicitly to be stored may decrease to zero, e.g. no data are available for special countries in the current year.
As entries are always 'packed' in SPEL, '=' has to be specified for the minimum number of data elements in an entry.

Maximum number of data elements in an entry

The maximum number means the highest number of data elements over all entries to be stored in the file.

The number specified has to be greater than, or equal to, the number actually occurring.
SPEL tables of basic table structure and complementary table structure are to be stored in TAB files. In SPEL the storing mode is always 'packed' (zero data are not explicitly stored). The number of data elements in the entries differs therefore.
Example:

There may be 2000 data elements estimated for maximum. However entering '3000' will create no disadvantages.

Estimated average number of data elements in an entry

See 'Minimum number of data elements in an entry'.
Enter your estimation of the average number of data elements explicitly stored, taking into account the total number of entries to be stored in the file.
Example:

The rough estimate may be about 450 data elements. To be on the safe side, enter e.g. '500'.

For information only:

Logical record length preferred

Length of each record in the TAB file in bytes.
The program computes a suitable record length. You can display the system's choice by pressing function key 5.
The logical record length entered is a multiple of 4 in the range 44 to 2048 (normal entries) or 48 to 2048 ('packed' entries).

Number of records to be reserved

The total number of records to be reserved for the TAB file.
The program computes a number suited to the specified number of entries and the average entry size. You can display the system's choice by pressing function key 5.

Example for creating a new file :

- In accordance with SPEL for the key-structure rules the total length will always be 16 characters. '16' must therefore be entered in the panel field '**Length of entry key in characters**'.

- The new files should have space for tables for 20 years, 15 regions and 10 table types, all in all 20 * 15 * 10 = 3000 tables. Additional entry space should be reserved for 'deleted' entries (See description of TAB files) and to allow storage of more tables in future. So '5000' may be entered in the panel field '**Maximum number of entries**'. Do not specify too few entries!

- The comment text to be stored with each entry may not exceed 80 characters (SPEL always reserves 80 characters for storing the date and time of the last modification). An '80' must be entered in the panel field '**Max. length of entry description in characters**'.

- SPEL tables of basic table structure and complementary table structure' tables are to be stored. In SPEL the storing mode is always 'packed'. The number of data elements in the entries therefore differ. In extreme cases the number of data elements explicitly to be stored (in SPEL the number of non-zero elements) will decrease to zero, e.g. no data are available for special countries in the current year. Entering a '0' in the panel field '**Minimum number of data elements in an entry**' will create no disadvantages.

- The estimate for the maximum number of data elements may be 2000. Entering a '3000' in the panel field '**Maximum number of data elements in an entry**' will create no disadvantages either.

- The rough estimate for the average number of data elements explicitly to be stored may be in the vicinity of 450. To be on the safe side, enter '500' in the panel field '**Estimated average number of data elements in an entry'.**

The figures entered in the panel fields 'Maximum number of entries' and 'Estimated average number of data elements in an entry' are the most important ones for determining the file space to be reserved. Overestimates only waste disk space. There is no need to input any of the 'Optional parameters'.

After the file has been successfully created, the table key structure should either be defined by the service 'Define key structure' or copied from an existing TAB file by the service 'Copy key structure'.

Define key structure:

This service lets you define a key structure which divides the table key into up to ten subkeys.

You will be prompted for a TAB file name and the subkey definitions. Up to ten subkeys may be defined, specifying name and length of each subkey.
The total of subkey lengths must be equal to the length of the whole key displayed at the top of the subkey definition panel.

Work file selection screen

```
Table file services -------------------- SPEL ----------------- Define key structure

                                Please enter file name




File to be processed (TAB) => SPEL-USR TAB A




                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

File to be processed (TAB)

The name of an existing TAB file for which the key structure is to be defined.

Key structure definition screen

```
Table file services -------------------- SPEL ---------------- Define key structure

          Please define the subkeys for a total key length of 16 characters




        Length and name of subkey:  1 =>  1 Region
        Length and name of subkey:  2 =>  2 Sub-region
        Length and name of subkey:  3 =>  2 Current year
        Length and name of subkey:  4 =>  2 Periodicity
        Length and name of subkey:  5 =>  2 Base year
        Length and name of subkey:  6 =>  4 Type
        Length and name of subkey:  7 =>  1 Model area
        Length and name of subkey:  8 =>
        Length and name of subkey:  9 =>
        Length and name of subkey: 10 =>




                Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Up to ten subkeys can be defined, specifying the name and length of each subkey. The total subkey length must be equal to the length of the whole key displayed at the top of the subkey definition panel.
According to SPEL rules, the total key length has to be 16 characters.
The codes for the subkey names are limited to a length of 20 characters.
For SPEL the following definitions have to be used:

```
Length and name of subkey:  1 =>  1 Region
Length and name of subkey:  2 =>  2 Sub-region
Length and name of subkey:  3 =>  2 Current year
Length and name of subkey:  4 =>  2 Periodicity
Length and name of subkey:  5 =>  2 Base year
Length and name of subkey:  6 =>  4 Type
Length and name of subkey:  7 =>  1 Model area
```

To remove an existing key structure, please define one subkey with the length of the whole key:

```
Length and name of subkey:  1 =>  16 Whole key
```

Delete entries:

This service lets you delete entries in a file.

You will be prompted for the TAB file name and for a key selection of entries to be deleted.

The service does not free the file space occupied by the deleted entries. You have to use the 'Free unused space' service to make the space reusable.

Work file selection screen

```
Table file services -------------------- SPEL ----------------------- Delete entries

                                Please enter file name




File to be processed (TAB) => SPEL-USR TAB A




                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

File to be processed (TAB)

The name of an existing TAB file for which entries are to be deleted.

Entry key selection screen

```
Table file services ---------------------- SPEL ----------------------- Delete entries

                              Please enter key selection



Region        ( 3 ch.) =>

Sub-region    ( 2 ch.) =>

Current year  ( 2 ch.) =>

Periodicity   ( 2 ch.) =>

Base year     ( 2 ch.) =>

Type          ( 4 ch.) =>

Model area    ( 1 ch.) =>


                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area, see 'Compare entries: - Entry key selection screen'.

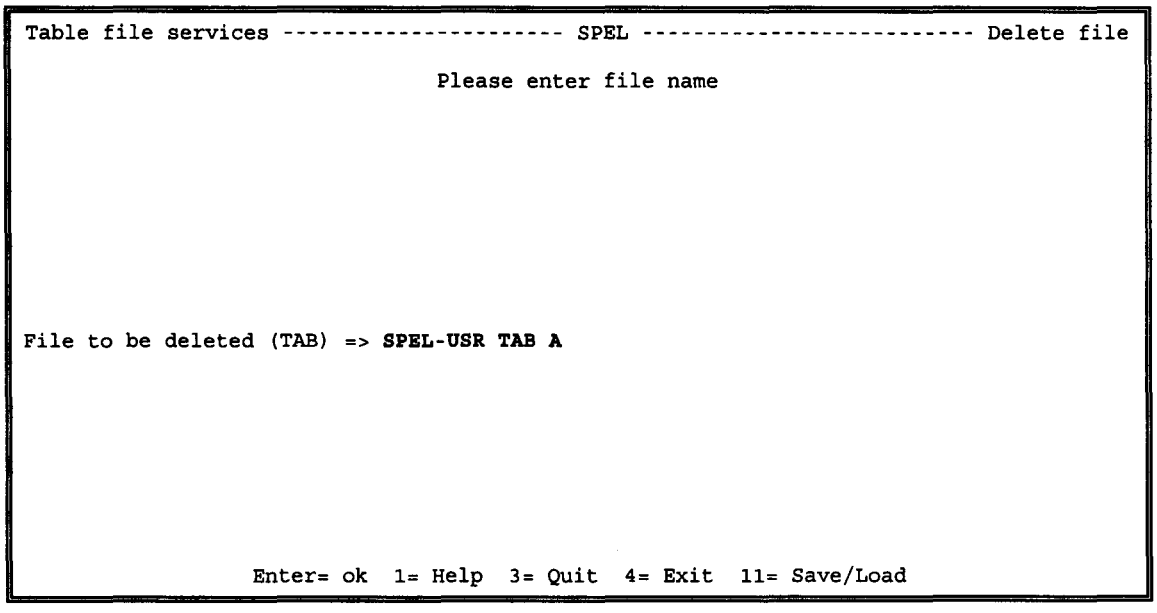
Remark : For maximum security no presets are shown in the screen above.

Delete file:

This service lets you delete an existing TAB file.

You will be prompted for the name of the TAB file to be deleted.

Work file selection screen



Parameters:

File to be deleted (TAB)

The name of an existing TAB file to be deleted.

Edit entry data:

This service lets you edit the data area of existing TAB file entries.

You will be prompted for the TAB file name and for a key selection of entries to be edited.

The edit screen may be moved horizontally and vertically over the data table by using function keys. Additional function keys let you page through the selected entries.

Work file selection screen

```
Table file services ---------------------- SPEL ---------------------- Edit entry data

                                Please enter file name




File to be processed (TAB) => SPEL-USR TAB A




                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

File to be processed (TAB)

The name of an existing TAB file to be edited.

Entry key selection screen

```
Table file services ---------- SPEL ---------- Edit entry data

                          Please enter key selection



Region       ( 3 ch.) => D F

Sub-region   ( 2 ch.) => 00

Current year ( 2 ch.) => 85 : 89

Periodicity  ( 2 ch.) => 00

Base year    ( 2 ch.) => NN

Type         ( 4 ch.) => EXPC

Model area   ( 1 ch.) => S


            Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area, see 'Compare entries: - Entry key selection screen'.

Note:

You cannot create new entries.

Edit panel

```
Table file services ---------- SPEL ---------- Edit table

TABLE:  D   00 80 00 00 CONC S

             13 ZP13     14 ZP14     15 ZP15     16 ZP16     17 PR01     18 PR02

 17 FLAX        3.00        8.00       15.00       14.00        0.00        0.0
 18 TOBA        0.00        0.00        0.00        0.00      751.94      100.0
 19 OIND        0.00        0.00        0.00        0.00        0.00        0.0
 20 CAUL      194.42       52.77      219.41        0.00       73.28      100.0
 21 TOMA        0.00        0.00        0.00        0.00      110.72      100.0
 22 OVEG        0.00        0.00        0.00        0.00        0.00        0.0
 23 APPL        0.00        0.00        0.00        0.00        0.00        0.0
 24 OFRU        0.00        0.00        0.00        0.00        0.00        0.0
 25 CITR        0.00        0.00        0.00        0.00        0.00        0.0
 26 TABO        0.00        0.00        0.00        0.00        0.00        0.0
 27 NURS        0.00        0.00        0.00        0.00        0.00        0.0
 28 FLOW        0.00        0.00        0.00        0.00        0.00        0.0
 29 OCRO        0.00        0.00        0.00        0.00        0.00        0.0
 30 TWIN        0.00        0.00        0.00        0.00        0.00        0.0
 31 OWIN        0.00        0.00        0.00        0.00        0.00        0.0
 32 OROO      205.73      102.87      401.18        1.00        0.00        0.0
          1= Help  2= Refresh  3= Quit  4= Exit  6= Bottom
        8= Forward  10= Right  11= Save/Load  12= Next Table
```

An edit panel is like a window which can be moved over the whole table. The second header line always shows the key of the current table. The next line displays the sequence numbers and codes for table columns in the current window. At the beginning of the subsequent lines you will find the sequence numbers and codes for table rows.

The data area at the intersection points of row and column codes may be modified.

Special function keys:

function key 2 : Re-establish the data as they were immediately after the last ENTER or function key processing.

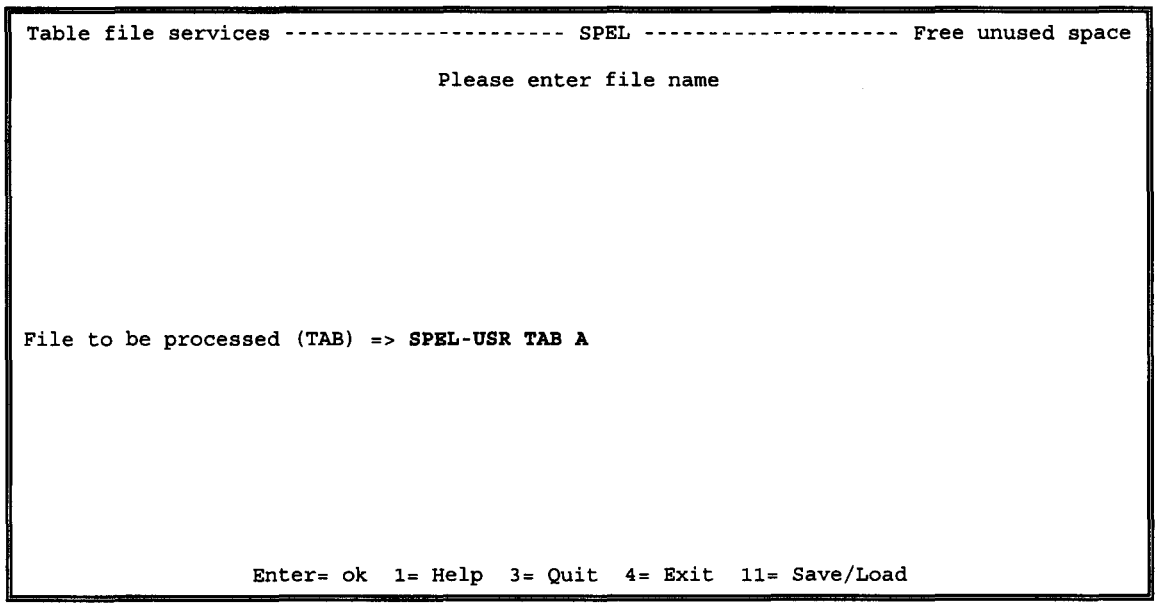
function key 5 to 10 : Move the screen window over the whole table.

function key 11/12 : Move to the previous/next table in the entry selection.

Free unused space:

This service lets you free unused space, i.e. file directory space and file space occupied by 'deleted' entries or entry parts.
You will be prompted for the name of the TAB file to be processed.
The checks of the 'Test file consistency' service are included in this service. If inconsistencies are found, the message 'Inconsistencies found' is displayed.
Inconsistencies may only occur as a result of program errors or abnormal program endings. All incorrect entries detected will be removed. The result is always a formally correct file, but there is no guarantee that the check algorithm can detect all entries affected by the abnormal program run.

Work file selection screen



Parameters:

File to be processed (TAB)

The name of an existing TAB file to be processed

List entry data:

This service lets you list the data area of TAB file entries. The service works like the service 'Edit entry data' without modification facilities.

List file directory:

This service lets you list the entry names (keys) and entry text of a selection of entries in a file.

You will prompted for the TAB file name and for a key selection.

Work file selection screen

```
Table file services ---------------------- SPEL ------------------ List file directory

                                Please enter file name









File to be processed (TAB) => SPEL-USR TAB A









                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

File to be processed (TAB)

The name of an existing TAB file for which the file directory is to be listed.

Entry key selection screen

```
Table file services ---------------------- SPEL ------------------ List file directory

                              Please enter key selection



Region         ( 3 ch.) => NL

Sub-region     ( 2 ch.) => 00

Current year   ( 2 ch.) => 60 : 99

Periodicity    ( 2 ch.) => 00

Base year      ( 2 ch.) => NN

Type           ( 4 ch.) => COMC

Model area     ( 1 ch.) => S


                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area, see 'Compare entries: - Entry key selection screen'.

Directory list screen

```
Table file services -------------------- SPEL ------------------ List file directory
 ------------------------------------------------------------------------------
 Key                              size  stored   text
 ------------------------------------------------------------------------------
 NL  00 80 00 00 COMC S    11040    1302   09.02.91 17:10:39
 NL  00 81 00 00 COMC S    11040    1302   09.02.91 17:10:39
 NL  00 82 00 00 COMC S    11040    1302   09.02.91 17:10:39
 NL  00 83 00 00 COMC S    11040    1302   09.02.91 17:10:39
 NL  00 84 00 00 COMC S    11040    1302   09.02.91 17:10:39
 NL  00 85 00 00 COMC S    11040    1302   09.02.91 17:10:39
 NL  00 86 00 00 COMC S    11040    1302   09.02.91 17:10:39
 NL  00 87 00 00 COMC S    11040    1302   09.02.91 17:10:39
 NL  00 88 00 00 COMC S    11040    1303   09.02.91 17:10:39
 NL  00 89 00 00 COMC S    11040    1301   09.02.91 17:10:39

 Total    10 entries listed




                              3= Quit  4= Find  12= Print
```

Parameters:

None.

Explanation:

- The 'size' column shows the number of elements in the table. In SPEL this number is equal to rows multiplied by columns.
- The 'stored' column shows the number of explicitly stored data elements. In SPEL it is the number of 'non-zero' elements in the table.
- The 'text' column shows comments stored with the entry. In SPEL it corresponds to the date and time of the last modification.
- If the file directory list does not fit one screen, there are additional function keys to move through the whole list.

Special function keys:

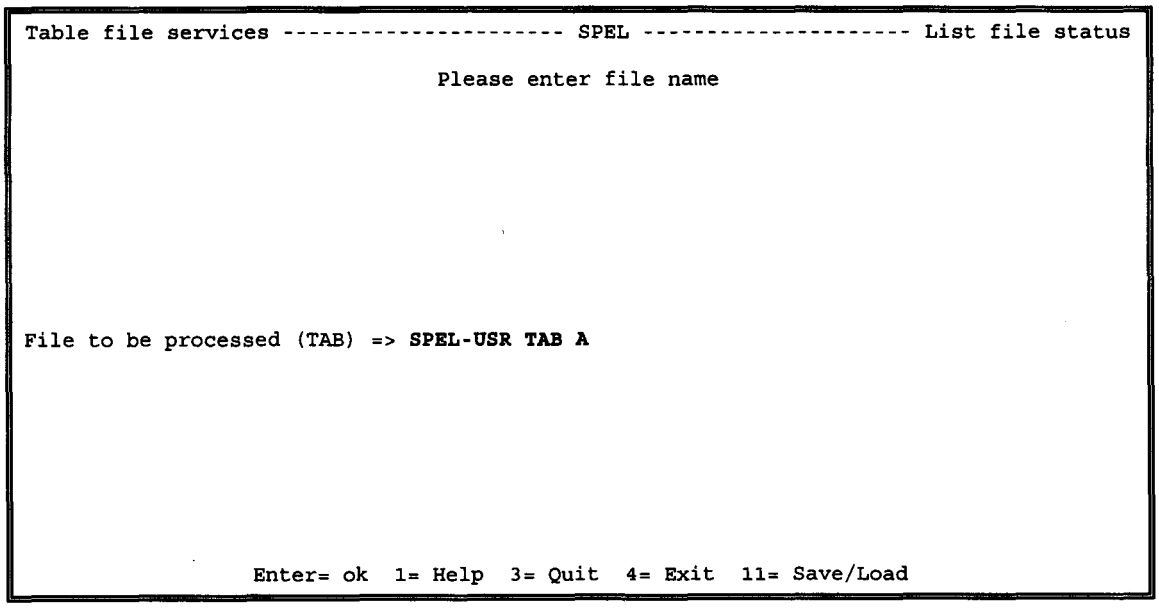
function key 4: Displays a special field to enter a search string to be searched.

List file status:

This service lets you list information about file characteristics, including capacity and usage information for the number of entries and records.

You will prompted for the TAB file name.

Work file selection screen



Parameters:

File to be processed (TAB)

The name of an existing TAB file for which the status is to be listed.

Status screen

```
Table file services -------------------- SPEL ---------------------- List file status

          S T A T U S   O F   T H E   D A O P E N   F I L E   S Y S T E M


Dir. space max.:   600000         active:    5760
        deleted:        0         free:    594240         free %:    99.0

Files in access:        1



File:           SPEL-USR TAB  A
Status, priority:       1         I/O:       READ
Length, record:      1024         key:         16

Entries max.:         512         active:      288
        deleted:        0         free:        224         free %:    43.8

Records max.:        6011         active:     4851
        deleted:        0         free:       1160         free %:    19.3


        3= Quit  4= Find  6= Bottom  8= Page down  11= 1/2 p.down  12= Print
```

Parameters:

None.

Explanation:

Files in access:

Always '1' in the 'List file status' service.

Directory space:

Program core memory in bytes reserved for the internal TAB file directory.

Status priority:

Priority in the list of open files (always 1).

Status I/O:

Input/output status (always 'READ').

Length, record:

Record length in bytes.

Length, key:

Key length in bytes.

Entries:

Table keys in the internal file directory.

Records:

Records in the file, the smallest internally addressable unit in the file.
An entry (='table') uses one or more records.

Directories Space / Entries / Record:

... max.:
Total reserved space, can never be exceeded.

... active:
Actually in use by accessible entries.

... deleted:
De-activated by implicit or explicit deletion of entries, can be made reusable by using the 'Free unused space' service.

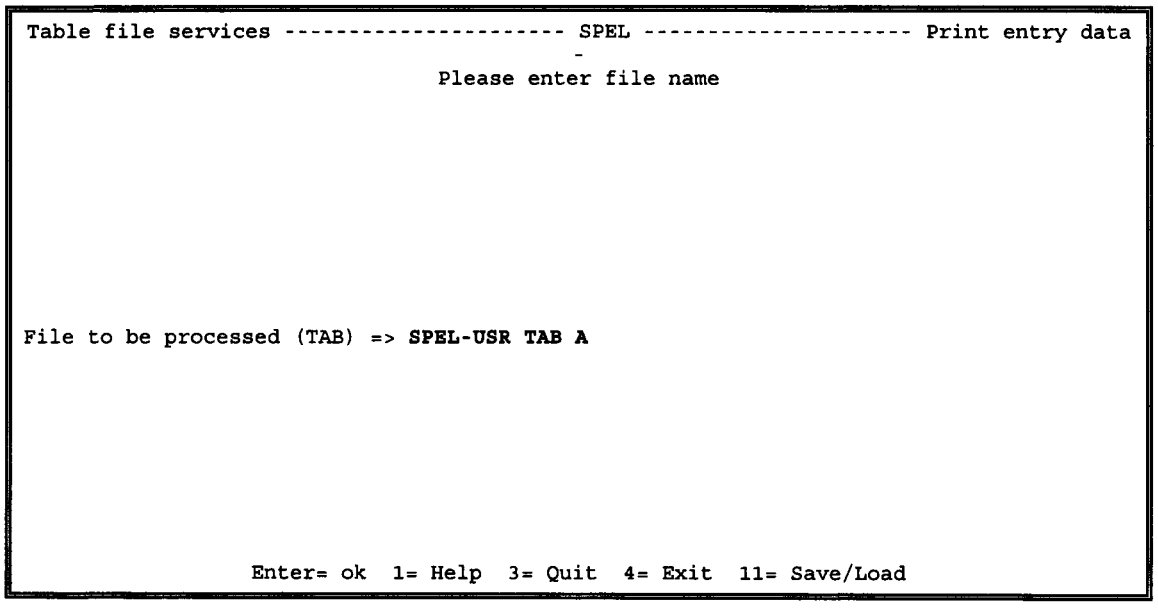
... free
Immediately usable for new entries.

Print entry data:

This service lets you print the data of TAB file entries as 'tables'.

You will be prompted for the TAB file name and for a key selection of entries to be printed.

Work file selection screen



Parameters:

File to be processed (TAB)

The name of an existing TAB file for which entries are to be printed.

Entry key selection screen

```
Table file services ---------------------- SPEL --------------------- Print entry data

                              Please enter key selection



Region        ( 3 ch.) => NL

Sub-region    ( 2 ch.) => 00

Current year  ( 2 ch.) => 60 : 99

Periodicity   ( 2 ch.) => 00

Base year     ( 2 ch.) => NN

Type          ( 4 ch.) => COMC

Model area    ( 1 ch.) => S


                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

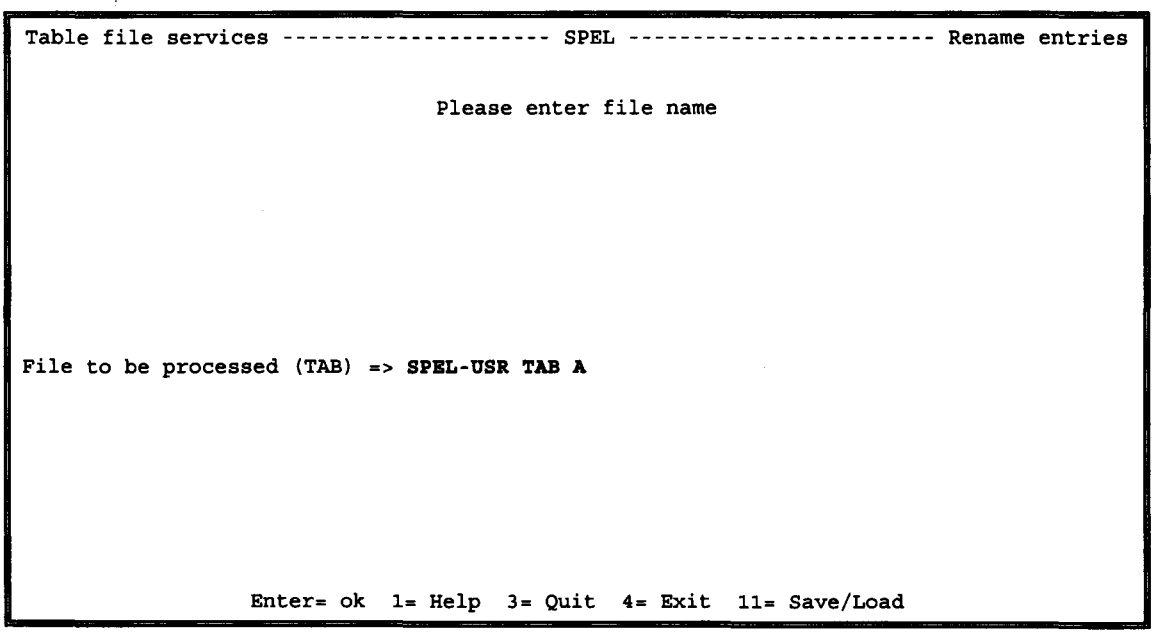
For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area, see 'Compare entries: - Entry key selection screen'.

Rename entries:

This service lets you rename entries in a TAB file. This means that the key will change without copying or moving any data.

You will be prompted for the TAB file name and for an entry selection with rename specifications.

Work file selection screen



Parameters:

File to be processed (TAB)

The name of an existing TAB file for which entries are to be renamed.

Entry key selection screen

```
Table file services ---------- SPEL ---------- Rename entries

                          Please enter key selection



Region        ( 3 ch.) => NL
              Rename =>
Sub-region    ( 2 ch.) => 00
              Rename =>
Current year  ( 2 ch.) => 60 : 88
              Rename =>
Periodicity   ( 2 ch.) => 00
              Rename =>
Base year     ( 2 ch.) => NN
              Rename =>
Type          ( 4 ch.) => COMC
              Rename => XXXC
Model area    ( 1 ch.) => S
              Rename =>


            Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

For parameter descriptions for region, sub-region, current year, periodicity, base year, type and model area, see 'Compare entries: - Entry key selection screen'.

rename

The subkeys to be renamed from the subkeys specified in the previous line have to be specified in this line.
You may specify one code or a sequence of codes for each subkey referring to the old subkeys. When using sequences, the n'th code in the sequence specifies the new code for the n'th reference code. The number of codes of both sequences therefore has to be equal. Ranges are not allowed. The selection line before has to contain the corresponding subkey codes.

If there is no input in this line, the same subkeys codes as specified in the line above are used for rename.

Test file consistency:

This service lets you test the file directory for inconsistencies which may only occur as a result of program errors or abnormal program endings.
If inconsistencies are found the message 'Directory wrong' is displayed. You can remove inconsistencies by using the service 'Free unused space', but you will lose data.
You will be prompted for the name of the TAB file to be processed.

Work file selection screen

```
Table file services ------------------- SPEL ----------------- Test file consistency

                                Please enter file name




File to be tested (TAB) => SPEL-USR TAB A




                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

File to be tested (TAB)

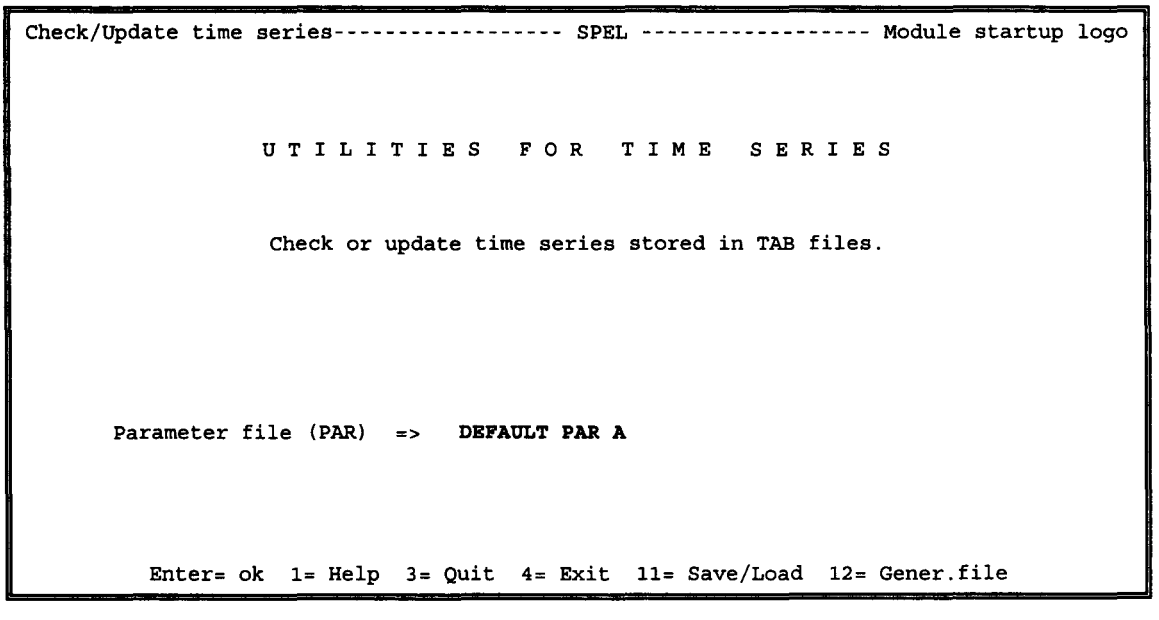
The name of an existing TAB file for which the consistency is to be checked.

6.1.2. TABUTL program: check time series

The TABUTL program offers the user three different services to check and compare time series stored in a SPEL work file of TAB format.

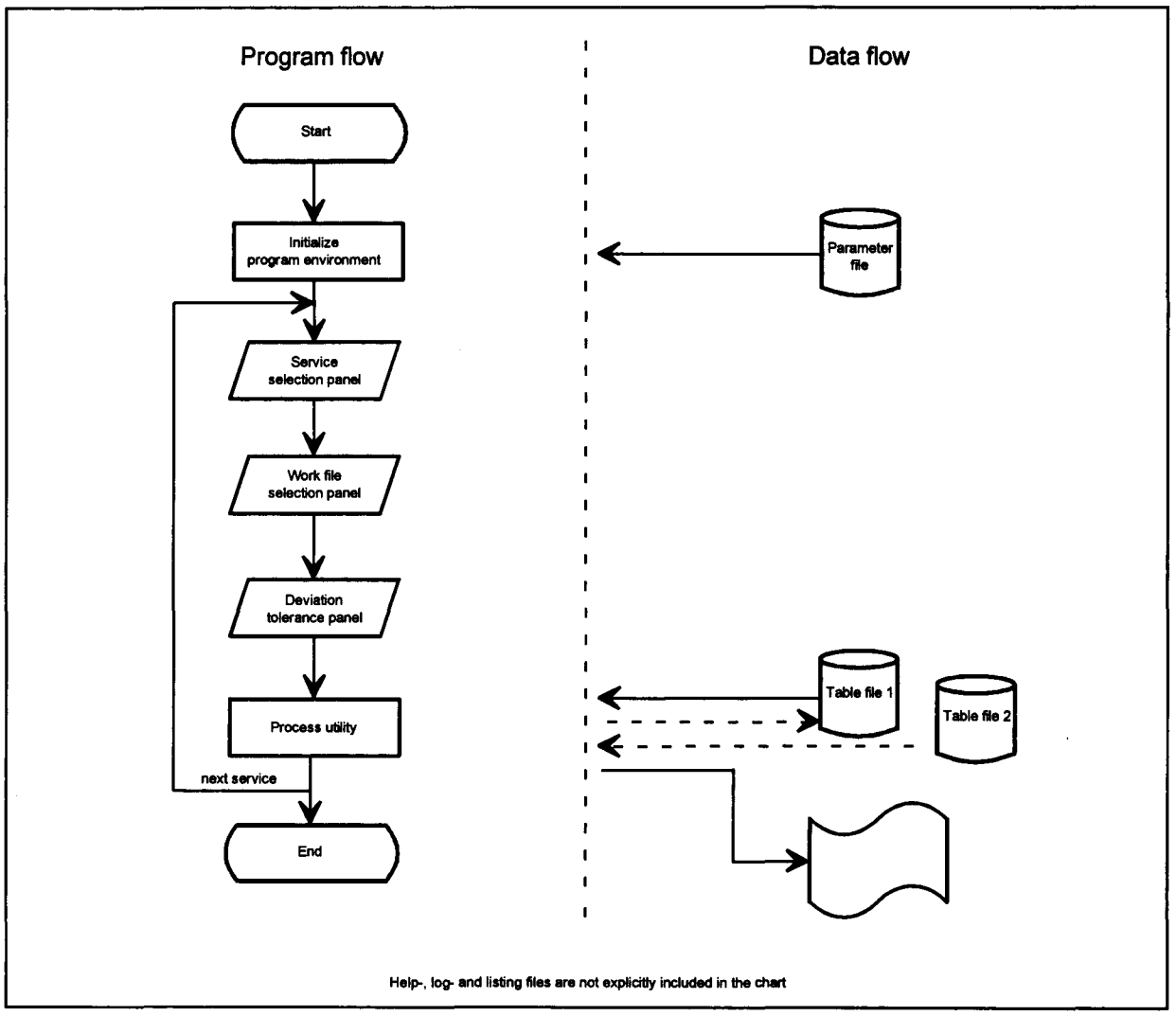
The user can define the tolerance for allowed deviations.

Module startup screen

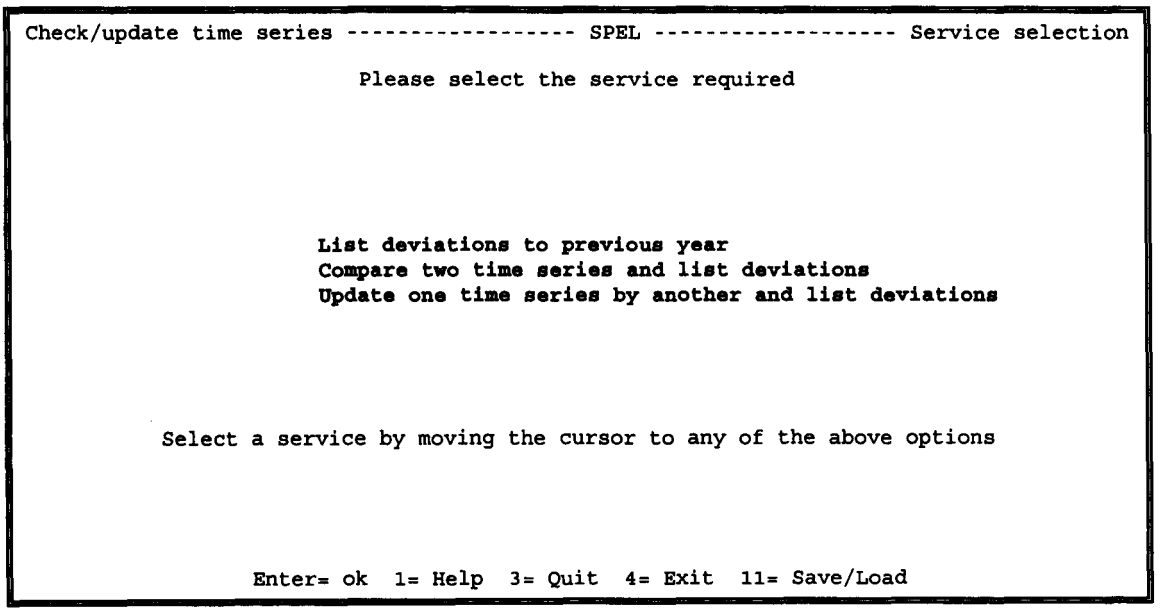


For detailed information, see chapter 'General programs'.

Figure 5: TABUTL program flow chart



Service selection screen



Selection items:

List deviations to previous year

The deviations of the current year data to the previous year data are listed.

Compare two time series and list deviations

The deviations of the current year data are listed.

Update one time series by another and list deviations

The deviations of the data are listed and all missing data within a specified update interval of the first time series are updated by the data of the second time series for the same year.

List deviations to previous year:

This service works on one time series. The deviations of the current year data to the previous year data are listed.

You will be prompted for the TAB file name and the PRN file name, for an entry key selection and the tolerance definitions for deviations.

Work file selection screen

```
Check/update time series------------------- SPEL ----------------- Work file selection

                              Please select file names




   Input file (TAB)   => SPEL-USR TAB A

   Print file (PRN)   => TABUTL PRN C




                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Input file (TAB)

This file is used for input of the reference time series. The file must be of TAB format.

Print file (PRN)

This file is used to list the data deviations.

Entry key selection screen

```
Check/update time series---------------- SPEL ------------------ Entry key selection

                              Please select table keys


Region        ( 3 ch.) => D F

Sub-region    ( 2 ch.) => 00

Periodicity   ( 2 ch.) => 00

Base year     ( 2 ch.) => NN

Type          ( 4 ch.) => EXPC

Model area    ( 1 ch.) => S

Check years   ( 2 ch.) => 73 : 92


                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Region

Region selection. Specify a single region code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*'.

Sub-region

Sub-region selection. Specify a single sub-region code or a sequence of codes.

The whole region has the '00'. Other sub-region codes are currently not used.

Periodicity

Periodicity selection. Specify a single periodicity code or a sequence of codes.

The whole year has the code '00'. Other periodicity codes are currently not used.

Base year

Base year code selection. Specify a single base year or a sequence of base years.

Type

Type selection. Specify a single type code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*'.

Model area

Select one model area codes, such as 'S':

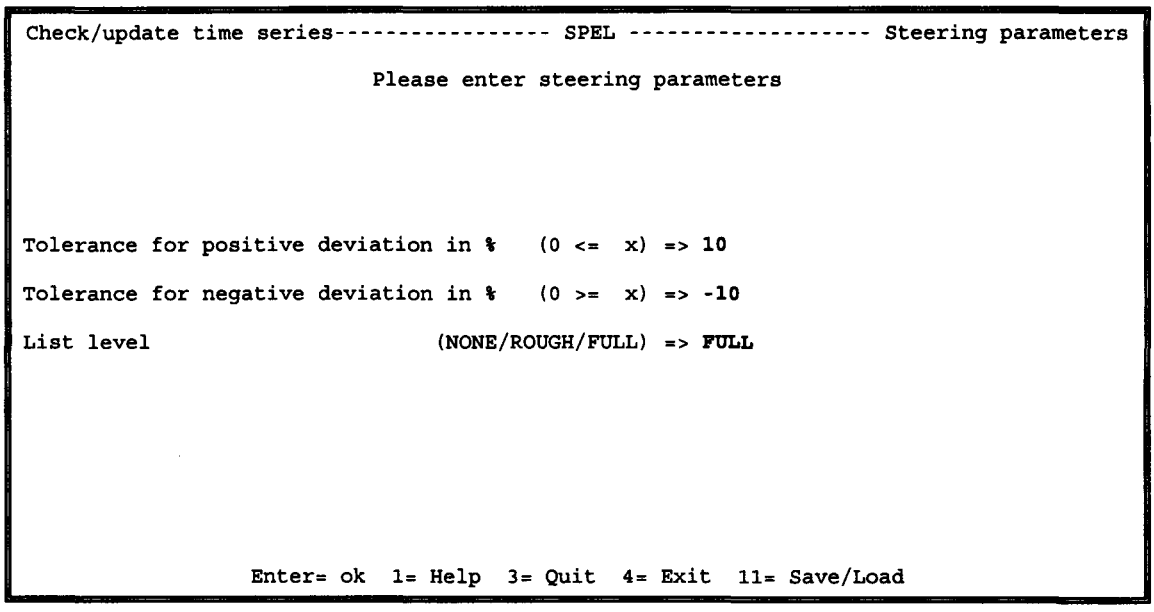
Check years

Years selection. Specify a single year, a sequence of years, a numerical range or a combination of sequences and ranges .

The maximum number of selectable codes for each parameter, except 'Model area', is 30.

For detailed information on code selection see chapter 'General program control'.

Steering parameter screen



Parameters:

Tolerance for positive deviation in %

Comparing data from the previous year and the current year, all deviations greater than 'x' % are listed in the LOG file.

Tolerance for negative deviation in %

Comparing data from the previous year and the current year all deviations lower than 'x' % are listed in the LOG file. For negative deviation the value 'x' is usually negative.

List level

NONE	No listing. Only the numbers of positive and negative deviations are listed.
ROUGH	The yearly deviation is listed.
FULL	The yearly data and the deviation are listed.

Compare two time series and list deviations:

This service works on two time series. The deviations of the current year data are listed. You will be prompted for the TAB file name, for an entry key selection and the tolerance definitions for deviations.

Work file selection screen

```
Check/update time series------------------- SPEL ----------------- Work file selection

                              Please select file names




   Reference file (TAB)=> SPEL-SYS TAB F

   Compare file (TAB)  => SPEL-USR TAB A

   Print file (PRN)    => TABUTL PRN C




                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Reference file (TAB)

This file is used for input of the reference time series. The file must be of TAB format.

Compare file (TAB)

This file is used for input of the compare time series and must be of TAB format.

Print file (PRN)

This file is used to list the data deviations.

Entry key selection screen

```
Check/update time series---------------- SPEL ------------------ Entry key selection

                     Please enter keys required to be compared



Region         ( 3 ch.) => D F
    Compare with        =>
Sub-region     ( 2 ch.) => 00
    Compare with        =>
Periodicity    ( 2 ch.) => 00
    Compare with        =>
Base year      ( 2 ch.) => NN
    Compare with        => 92
Type           ( 4 ch.) => BASB
    Compare with        => SFSB
Model area     ( 1 ch.) => S
Check years    ( 2 ch.) => 93




                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

For parameter descriptions for region, sub-region, periodicity, base year, type, model area and check year see 'List deviations to previous year: - Entry key selection screen'.

compare with

> The subkeys to be compared with the subkeys specified in the previous line have to be specified in this line.
>
> You may specify one code or a sequence of codes for each subkey referring to the compare file. When using sequences the n'th code in the sequence specifies the compare code for the n'th reference code. The number of codes of both sequences therefore has to be equal. Ranges are not allowed. The selection line before has to contain the corresponding subkey codes referring to the reference file.
>
> If there is no input in this line, the same subkeys as specified in the line above are used for comparison.

Steering parameter screen

```
Check/update time series---------------- SPEL ------------------ Steering parameters

                          Please enter steering parameters




Tolerance for positive deviation in %  (0  <=  x) => 10
Tolerance for negative deviation in %  (0  >=  x) => -10
List level                        (NONE/ROUGH/FULL) => ROUGH









                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Tolerance for positive deviation in %

When comparing data from the current year of the reference data and the compare data, all deviations greater than 'x' % are listed in the log file.

Tolerance for negative deviation in %

When comparing data from the current year of the reference data and the compare data, all deviations lower than 'x' % are listed in the log file.

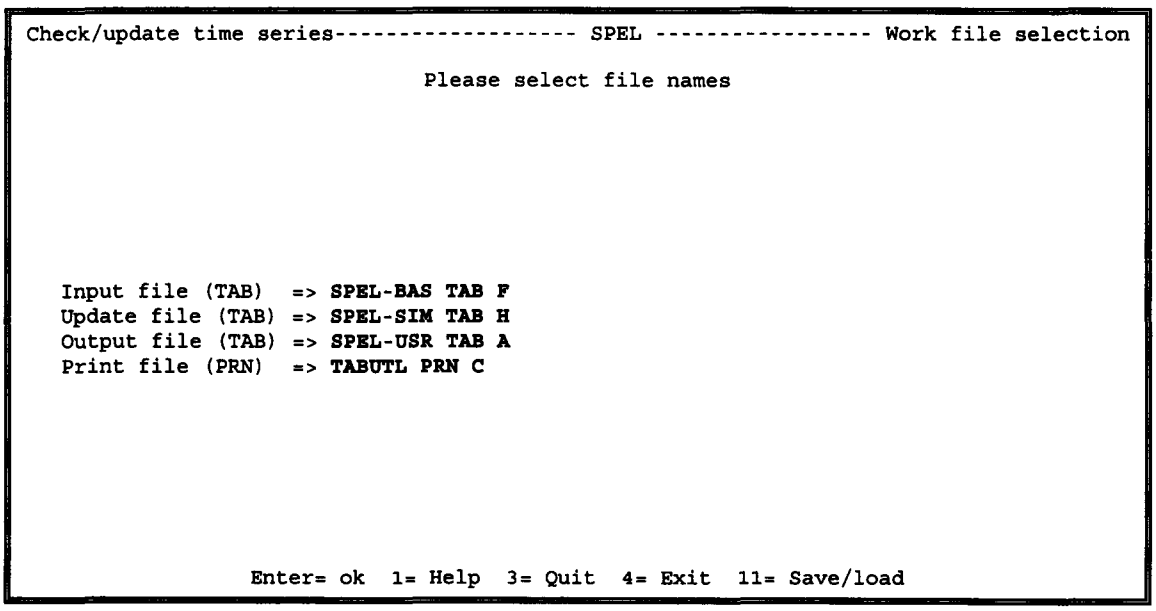
List level

NONE	No listing. Only the numbers of positive and negative deviations are listed.
ROUGH	The deviations of reference and compare data are listed.
FULL	The reference time series, the compare time series and the deviations are listed.

Update one time series by another and list deviations:

This service works on two time series. The deviations of the current year data are listed and all missing data within a specified update interval of the first time series are updated by the data of the second time series for the same year.
You will be prompted for the TAB file names, for an entry key selection and the tolerance definitions for deviations.

Work file selection screen



Parameters:

Input file (TAB)

This file is used for input of the reference time series. The file must be of TAB format. If there are 'data missings' within the time series, the 'missings' are updated by the data of the secondary input file.

Update file (TAB)

This file is used for input of the update time series and must be of TAB format.

Output file (TAB)

This file is used for output of the updated time series and must be of TAB format.

Print file (PRN)

This file is used to list the data deviations.

Entry key selection screen

```
 Check/update time series----------------- SPEL ------------------- Comparison keys

                     Please enter keys required to be compared


Region         ( 3 ch.) => D F
       Update by       =>
 Sub-region    ( 2 ch.) => 00
       Update by       =>
Periodicity    ( 2 ch.) => 00
       Update by       =>
Base year      ( 2 ch.) => NN
       Update by       => 92
Type           ( 4 ch.) => BASB
       Update by       => SFSB
Model area     ( 1 ch.) => S
Update years   ( 2 ch.) => 93
Output years   ( 2 ch.) => 72 : 93
Output type    ( 4 ch.) => SPEB


                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

For parameter descriptions for region, sub-region, periodicity, base year, type and model area see 'List deviations to previous year: - Entry key selection screen'.

Update years

> Years to be updated while checking reference and update time series. Specify a single year, a sequence of years, a numerical range or a combination of sequences and ranges.

Output years

> Years selection for output. Specify a single year, a sequence of years, a numerical range or a combination of sequences and ranges.

Output type

> Type selection. Specify a sequence of type codes, such as 'OU1B OU2B'.
>
> The number of output types must be equal to the number of input types. The table structure of each output type must fit with the table structure of the corresponding input type.

The maximum number of years and types that may be selected is 30.

update by ...

> The subkeys to be updated by the subkeys specified in the previous line of input codes have to be specified in this line.

You may specify one code or a sequence of codes for each subkey referring to the update file. When using sequences, the n'th code in the sequence specifies the update code for the n'th input code. The number of codes of both sequences therefore has to be equal. Ranges are not allowed. The selection line before has to contain the corresponding subkey codes referring to the input file.

If there is no input in this line, the same subkeys as specified in the line above are used for comparison.

Steering parameter screen

```
Check/update time series--------------- SPEL ----------------- Steering parameters

                          Please enter steering parameters




Tolerance for positive deviation in %  ( 0  <=  x) => 10
Tolerance for negative deviation in %   (0  >=  x) => -10
List level                          (NONE/ROUGH/FULL) => FULL
Output level                             (NONE/STORE) => STORE




                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Tolerance for positive deviation in %

When comparing data from the current year of the reference data and the update data, all deviations greater than 'x' % are listed in the protocol file.

Tolerance for negative deviation in %

When comparing data from the current year of the reference data and the compare data, all deviations lower than 'x' % are listed in the protocol file.

List level

NONE	No listing. Only the number of positive and negative deviations and the number of updates are listed.
ROUGH	The resulting time series are listed.
FULL	The reference time series, the compare time series, the deviations and the resulting time series are listed.

Output level

NONE	No storing.
STORE	The resulting time series are stored.

6.2. Data import and Export

6.2.1. ORIGIN program: importing standard sequential files into TAB files

Whenever data from sequential files are to be imported into tables in a TAB file, the ORIGIN program is used. A selection of input data is presented for each differentiation criterion (e.g. region, year, etc.). The user may define selection ranges.

Up to eight IMP and/or SDA input files are supported. If all input files are of the SDA format, it is not necessary to specify assignments to SPEL table element codes. If at least one of the input files is of the IMP format, an additional ASS assignment file of must be allocated.

An ASS control files selects thousands of source data codes and assigns them to SPEL table element codes. The assignments are specified only once for each table structure and are implicitly used for all tables matching the general selections (regions, sub-regions ...). Only one assignment file can be defined for all IMP input files. The set of all assignment statements in the ASS file illustrates the unambiguous relationship between original source data and SPEL table elements.

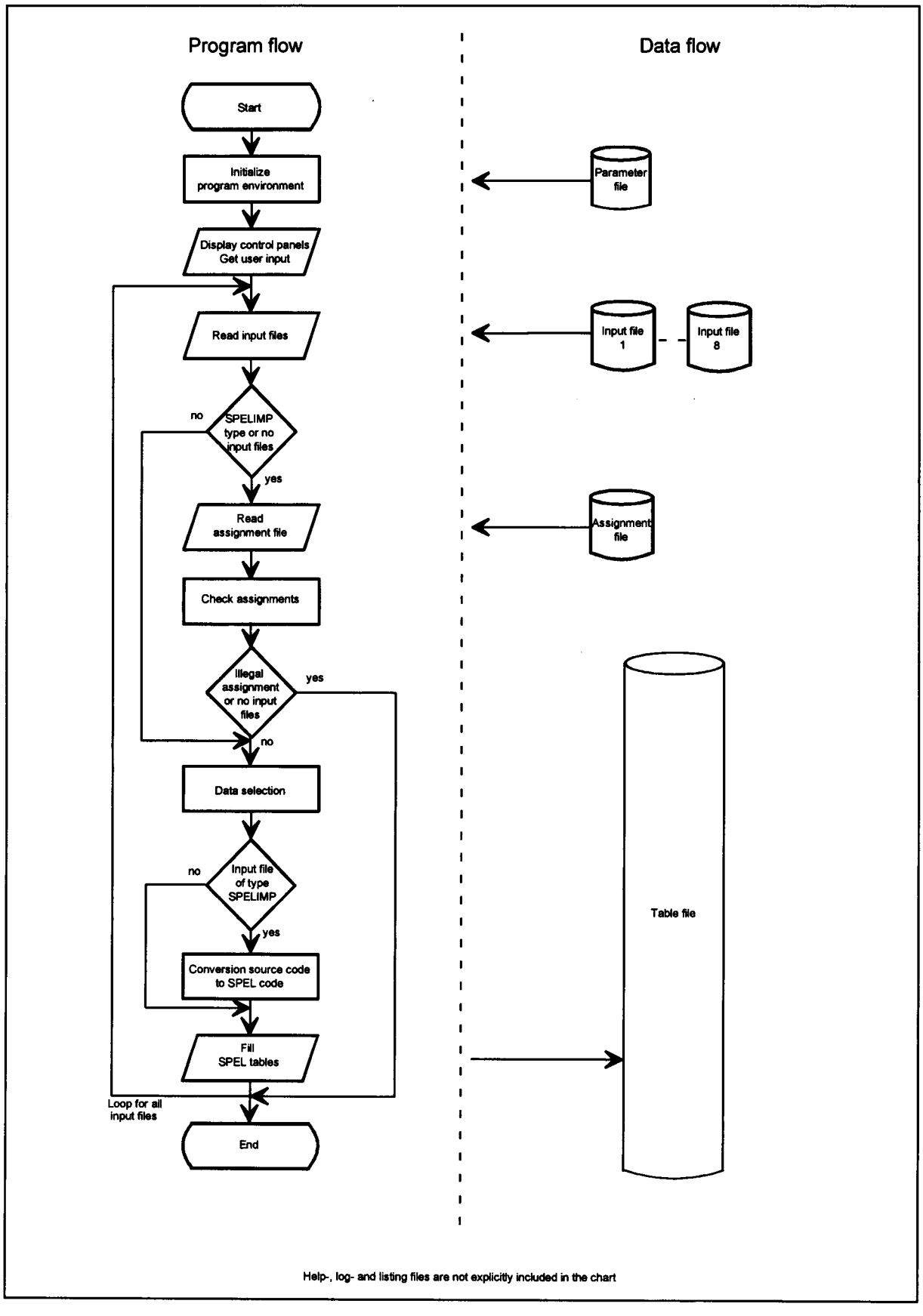
ORIGIN can also be used to check an assignment file for legal SPEL table element codes. In this mode the program does not process any data.

Module startup screen

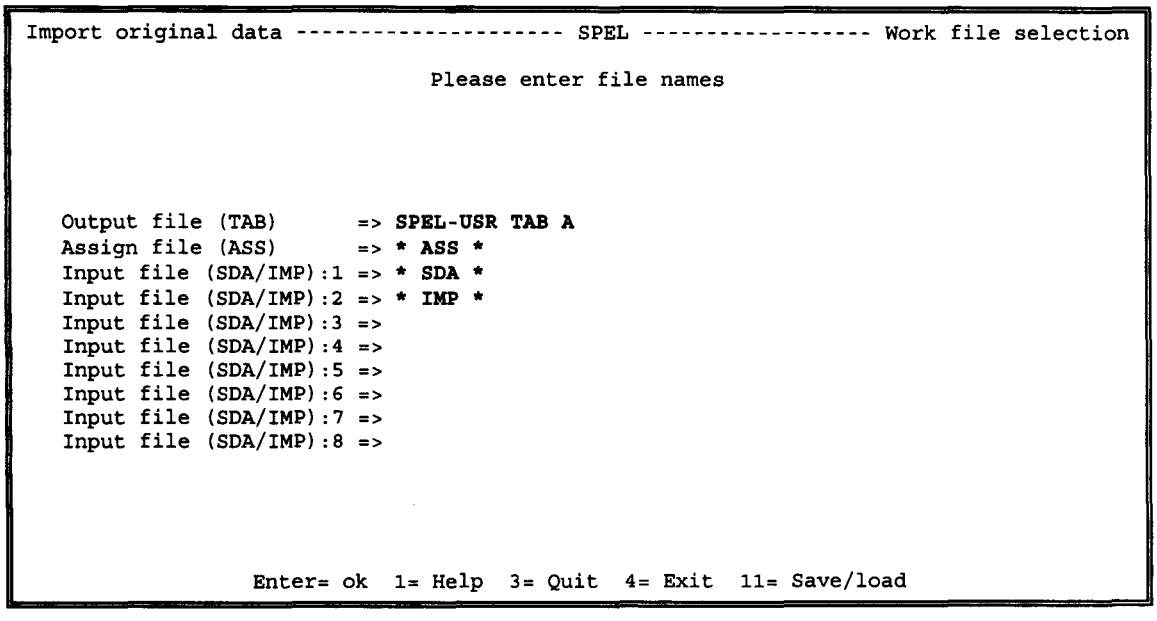
```
Import original data ------------------ SPEL -------------------- Module startup logo


        I M P O R T   D A T A   I N T O   S Y S T E M   D A T A   B A S E


              Get original data from files of SDA or IMP format,
          select data items with ASS files (if original format is IMP),
                        select table keys by dialog input,
                    store tables in a work file of TAB format.


     Parameter file (PAR)  =>   DEFAULT PAR A


        Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load  12= Gener.file
```

For detailed information, see chapter 'General programs'.

Figure 6: ORIGIN program flow chart



Work file selection screen



Parameters:

Output file (TAB)

This file is used for output and must be of TAB format.

All data which fit the selection criteria (next panel) will be stored as 'table' entries in this file.

Assign file (ASS)

This file contains the assignments for selecting source data and converting source data codes to SPEL table element codes.

The ASS file must be used if at least one of the input files is of the IMP format. Only one ASS file can be defined for all IMP input files.

Input file (SDA/IMP):1 ... 8

File of IMP or SDA one of the following two formats. The program identifies the file type automatically by analyzing the file structure.

Remark: If only the ASS file name is specified, ORIGIN will simply check the set of assignment statements for legal SPEL table element codes.

Table key selection screen

```
Import original data -------------------- SPEL ----------------- Table key selection

                Please make your key selection for output table file
                      Subkey for base year cannot be modified


Region          ( 3 ch.) => B : UK

Sub-region      ( 2 ch.) => 00

Current year    ( 2 ch.) => 72 : 93

Periodicity     ( 2 ch.) => 00

Base year       ( 2 ch.) => NN

Type            ( 4 ch.) => ZPAC

Model area      ( 1 ch.) => S


                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Region

Region selection. Specify a single region code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*'.

The maximum number of regions that may be selected is 400.

Sub-region

Sub-regions are currently not used.

The whole region has the sub-region code '00'. Always specify '00'.

Current year

Current year selection. Specify a single year, a sequence of years, a numerical range or a combination of sequences and ranges.

The maximum number of years that may be selected is 50.

Periodicity

Periodicity selection. Specify a single periodicity code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*' .

The whole year has the code '00'. Other periodicity codes are currently not used.

Base year

For putting original data into the table file the code for the subkey base year will always be 'NN'.

The subkey cannot be modified.

Type

Type selection. Specify a single type code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*'.

The maximum number of types that may be selected is 10.

Model area

Select one of the following model area codes:

'S' for supply
'D' for demand
'E' for supply & demand
'T' for trade

For detailed information on code selection see chapter 'General program control'.

Selection check screen

```
Import original data -------------------- SPEL --------------------- Selection check

 A T T E N T I O N: Make sure that you have specified all input files which are
 necessary to fill the tables for the specified types. If you have not there will
 be 'data missings' in the tables. At the worst they will contain only zeros, even
 if those tables have been completed before. Please check the current selection.

Types to be created: ZPAC

Input files        : CORR     SDA    A









              Select an action by moving the cursor to any topic below

                     Continue doing the selection
                     Revise selections for types/input files

                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

The user may once more reflect if all input files which are necessary to fill the tables for the selected types have been specified. If you have not, there will be 'data missings' in the tables. At the worst, they will contain only zeros, even if those tables have been completed before.

Selection items :

Continue doing the selection

Program execution continues selecting the input data for specified types.

Revise selection for types/input files

Program returns to the 'Table key selection'/'Work file selection' panel for new selections.

6.2.1.1. ASS files

ASS assignment files were created for selecting thousands of source data codes and assigning them to SPEL table element codes. The assignments are specified only once for each table structure and are implicitly used for all tables matching the general selections (regions, sub-regions ...).

The ASS files are sequential files with a fixed record length of 255. For user friendly editing, the line length should not exceed 72 characters.

File structure :

There are two record types :

- Comment records

Beginning with an '*' at the first character position. Comments may occur anywhere in the file.

- ASS statements

Each statement assigns an external data code to a SPEL table element code, respecting the model area and the table structure type.

Statement syntax:

ASS TARGET=*ms.cccc.rrrr,* SOURCE=*ttttddddddddd* [,SCALE=*ffffffff*];

Each statement must begin with the keyword ASS and must end with the delimiter ';'. The parameter values may be enclosed by quotation marks.

Parameter TARGET:

ms	Combination of model area code and table structure type
	m model area code such as 'S'
	s 'C' for complementary structure or 'B' for basic structure
cccc	SPEL table column code
rrrr	SPEL table row code

Parameter SOURCE:

tttt	Source type code (domain)
ddddddddd	Source data code, up to 9 characters (See IMP files)

Parameter SCALE:

ffffffff Scale factor for all data belonging to this code, when the data are assigned to SPEL table element codes.
If the parameter is missing, SCALE=1. is assumed.

The remaining text following the delimiter is treated as a comment and ignored.

Example of an ASS file:

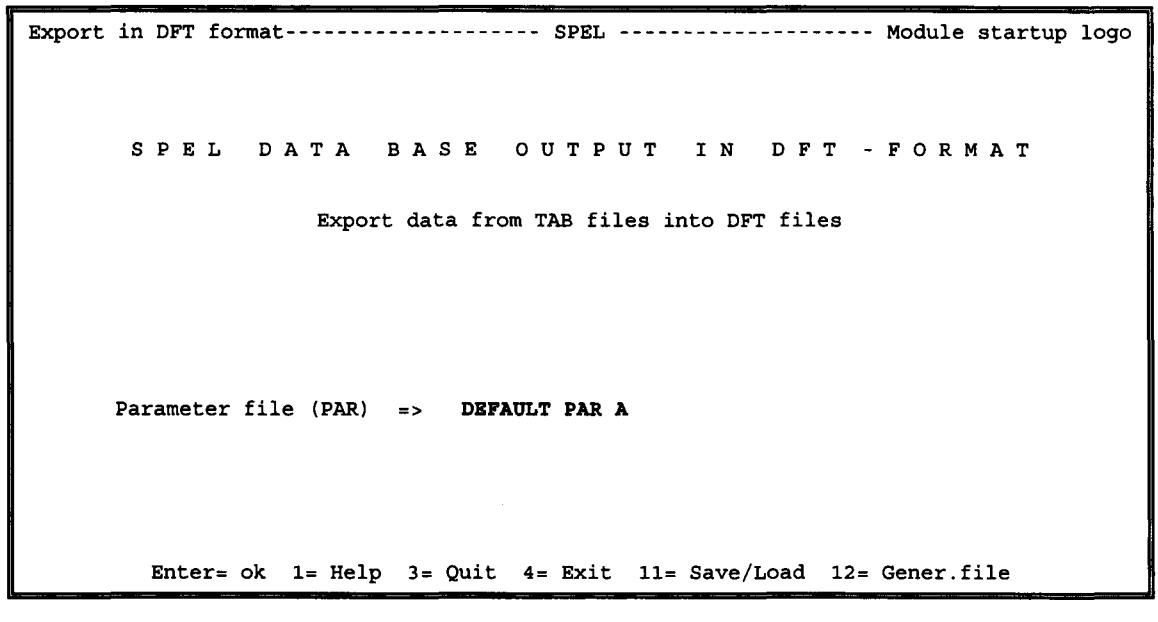
```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
*Example
ASS TARGET=SC.LEVL.SWHE, SOURCE=ZPA13112020 ;
ASS TARGET=SC.PR01.SWHE, SOURCE=PRAG3221120, SCALE=10.;
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
```

6.2.2. DADFT program: export data in DFT format for dissemination

For dissemination of the SPEL/EU-Data, EUROSTAT distributes the PC software package CUB.X. For CUB.X all input data have to be stored in DFT format. The user may call the DADFT program to define a data selection stored in tables of a file of TAB format to convert to DFT format. The program supports two versions of DFT formats:

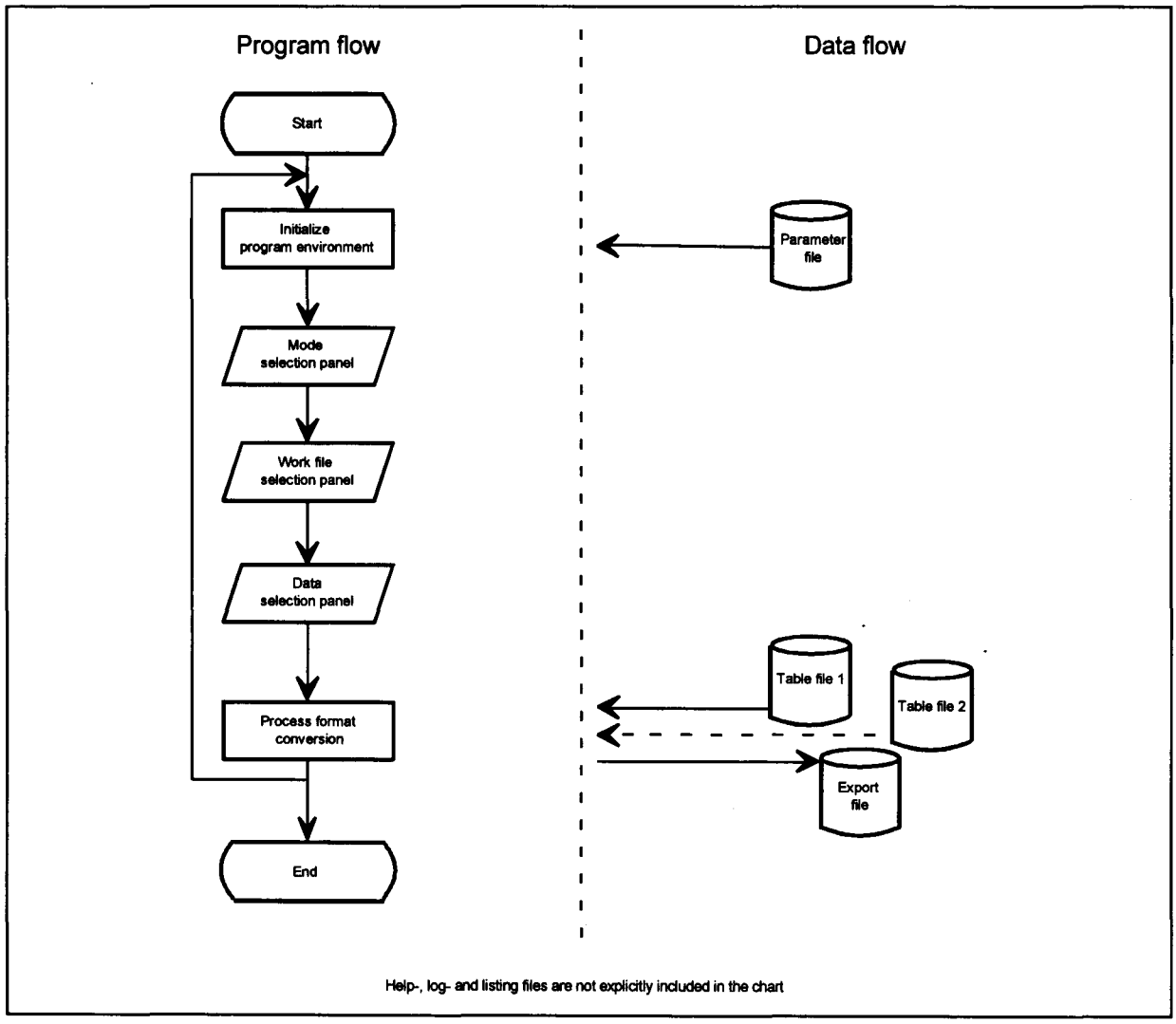
- a DFT format version for CUB.X running under DOS
- a DFT format version for CUB.X running under (DOS/) WINDOWS.

Module startup screen

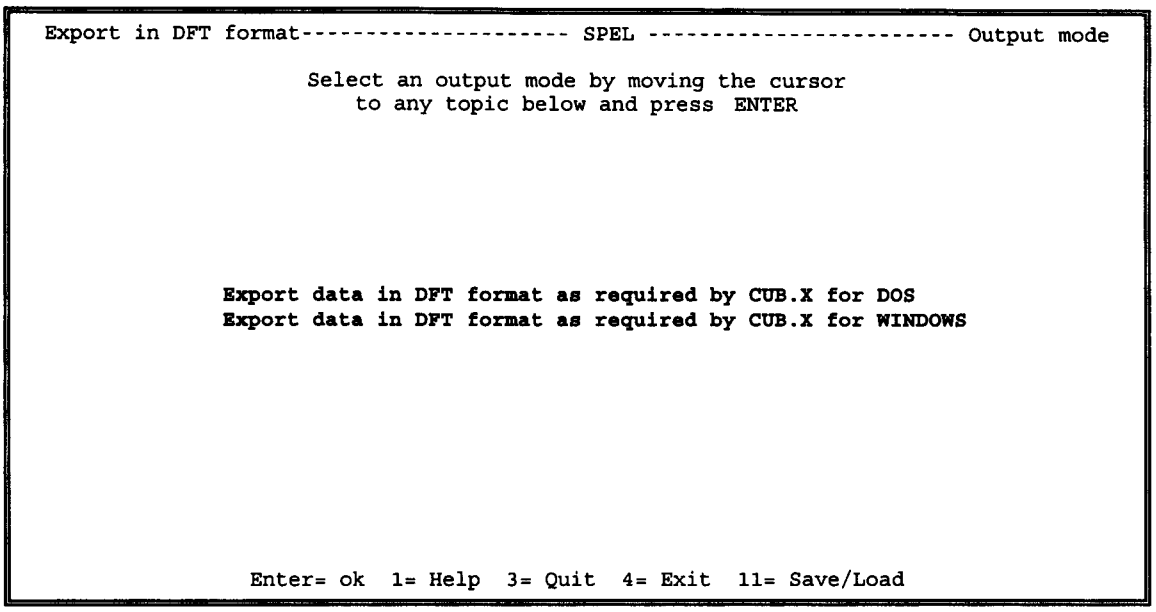


For detailed information, see chapter 'General programs'.

Figure 7: DADFT program flow chart



Output mode screen



Selection items:

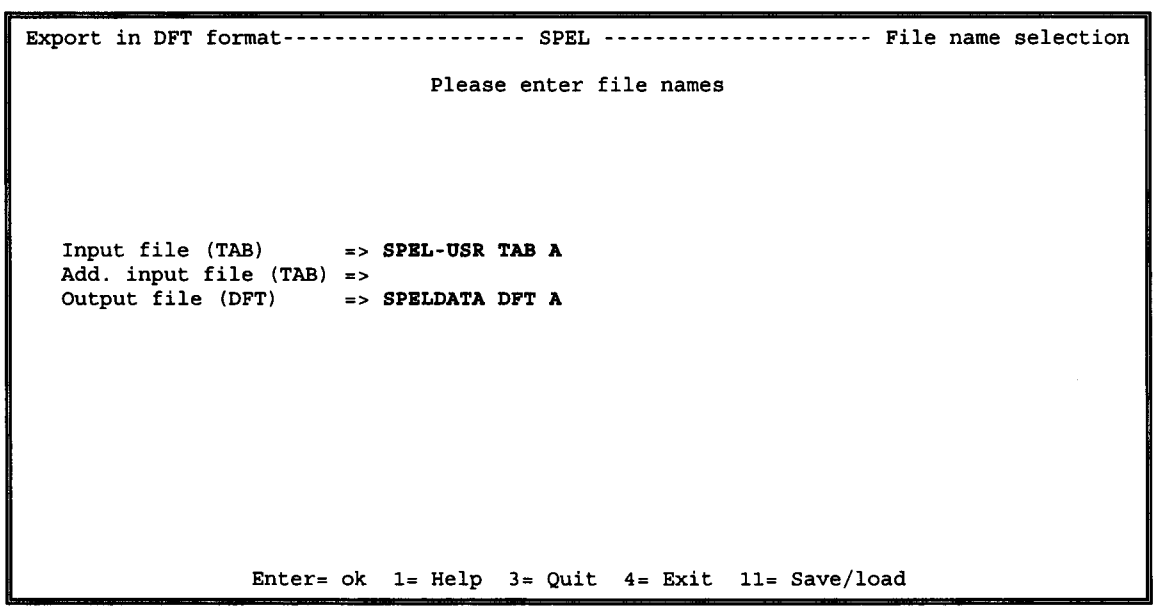
Export data in DFT format as required by CUB.X for DOS

Data stored in tables of a file of TAB format is converted into DFT format as required by CUB.X for DOS.

Export data in DFT format as required by CUB.X for WINDOWS

Data stored in tables of a file of TAB format is converted into DFT format as required by CUB.X for WINDOWS.

Work file selection screen



Parameters:

Input file (TAB)

This file is used for data input. The file must be of TAB format.

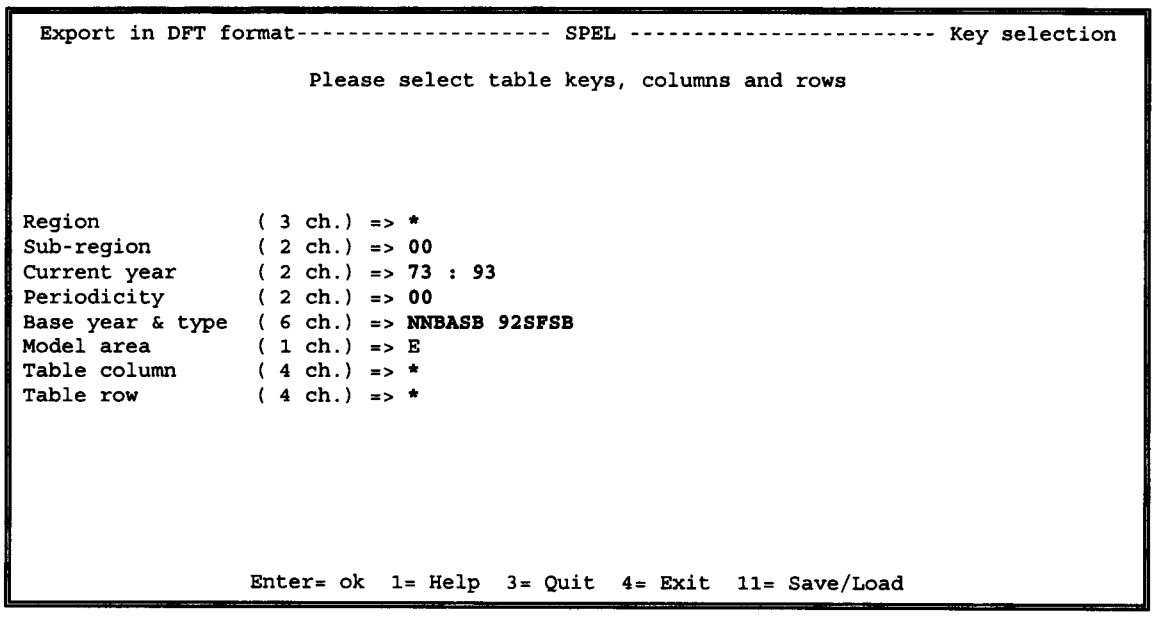
Add. input file (TAB)

Additional input file of TAB format. This file is to be accessed for tables not found in the first defined TAB file.

Output file (DFT)

Output file of DFT format version as selected by output mode.

Key selection screen



Parameters:

Region

Region selection. Specify a single region code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

Sub-region

Sub-region selection. Specify a single sub-region code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

Sub-regions are currently not used. The whole region has the sub-region code '00'.

Current year

Year selection. Specify a single year, a sequence of years, a numerical range, a combination of sequences and ranges .

Periodicity

Periodicity selection. Specify a single periodicity code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

Periodicities are currently not used. The whole year has the code '00'.

Base year & type

Base year and table type selection. Specify one combination or a sequence of combinations of base year and table type code such as 'NNBASB 91SFSB'.

Model area

The model area code cannot be modified and is always 'E'.

Table column

Table column selection. Specify a single column code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*' .

Table row

Table row selection. Specify a single row code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*' .

For detailed information on code selection see chapter 'General program control'.

6.2.3. DAOUT program: Print and export data from TAB file

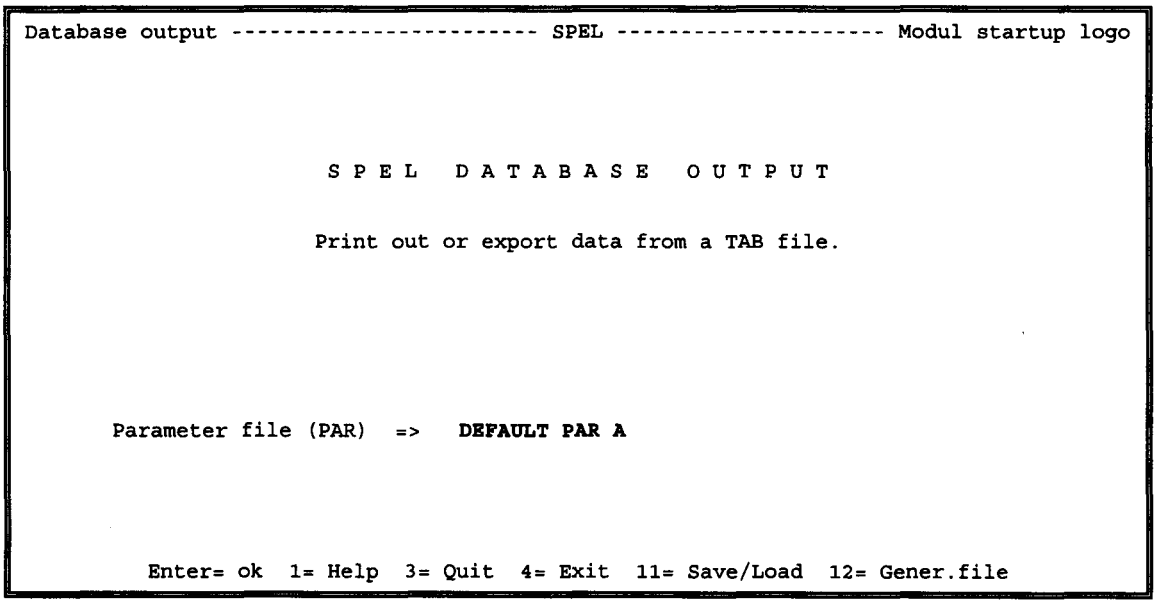
Selections of data from TAB files can be printed and exported by the DAOUT utility. The data selection is done by specifying the identifiers for the SPEL data dimensions (region, sub-region, year, periodicity, base year[6], table type, table column and table row). The data dimensions can be transposed for output, except for SDA output modus.

There are four different output modi:

- printing file
- SDA file
- file of CSV format for import to PC standard software, e.g. EXCEL
- file of German CSV format (';' as column separator)

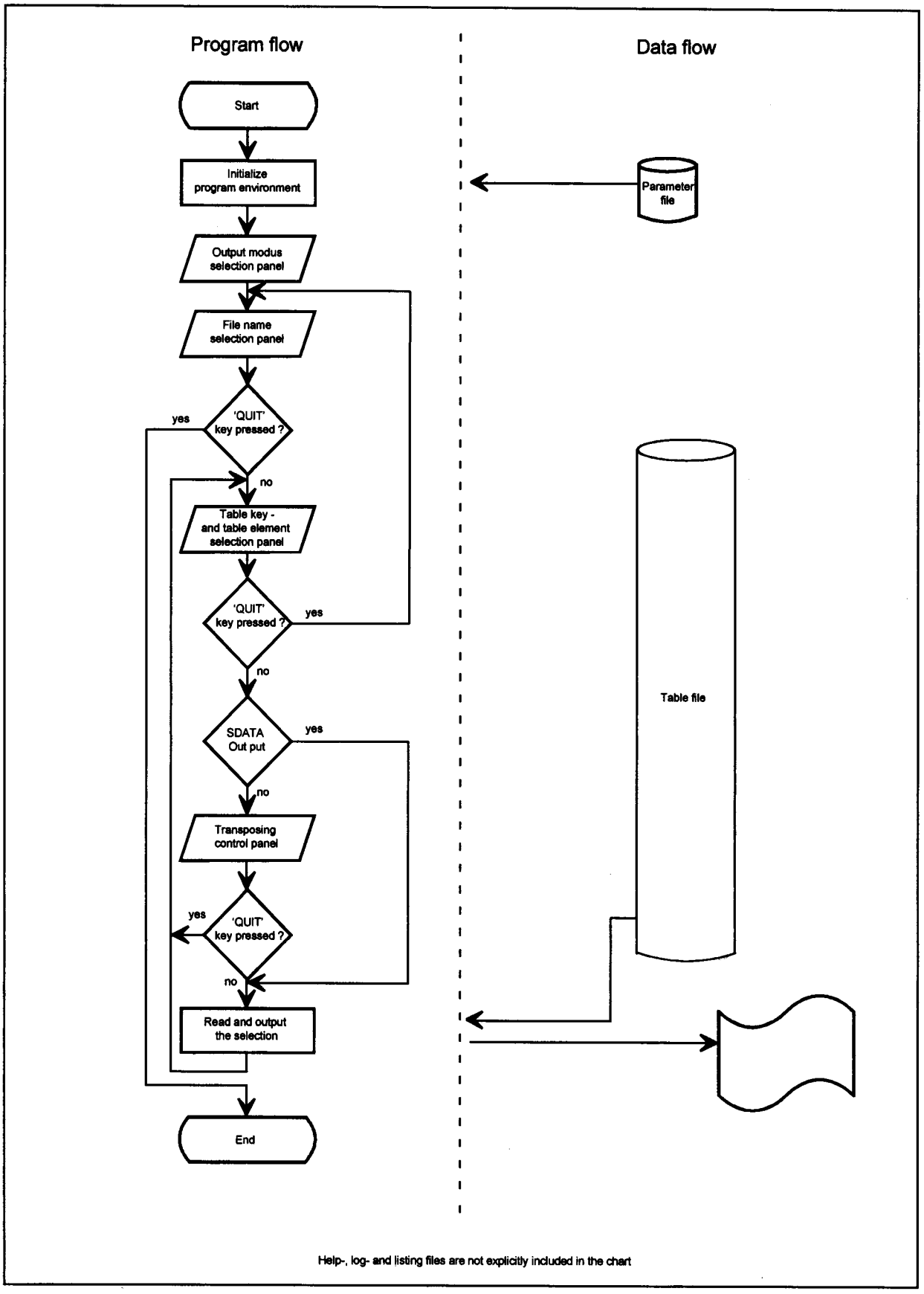
6 Currently base year and table type are combined to the SPEL data dimension "status". See Figure 2: SPEL-Data dimensions

Module startup screen

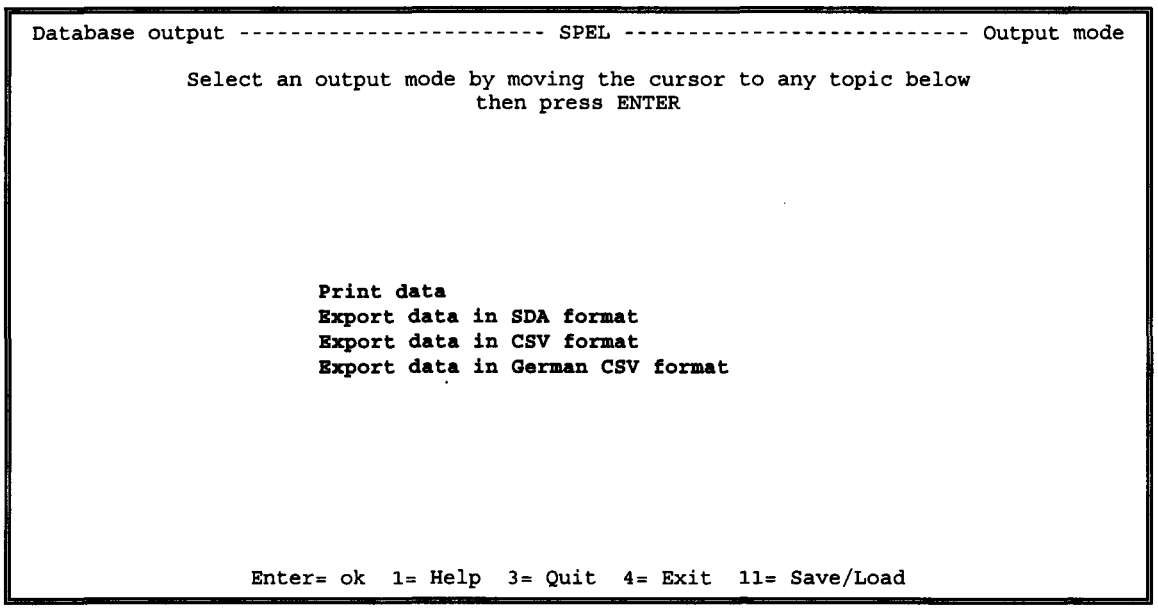


For detailed information, see chapter 'General programs'.

Figure 8: DAOUT program flow chart



Output mode selection screen



The output mode can be preselected by the 'Output mode' parameter in the parameter file.

Parameters:

Print data

This mode provides printing of data in the PRN file.

Export data in SDA format

The selected data from the TAB file are stored in a file of SDA format.

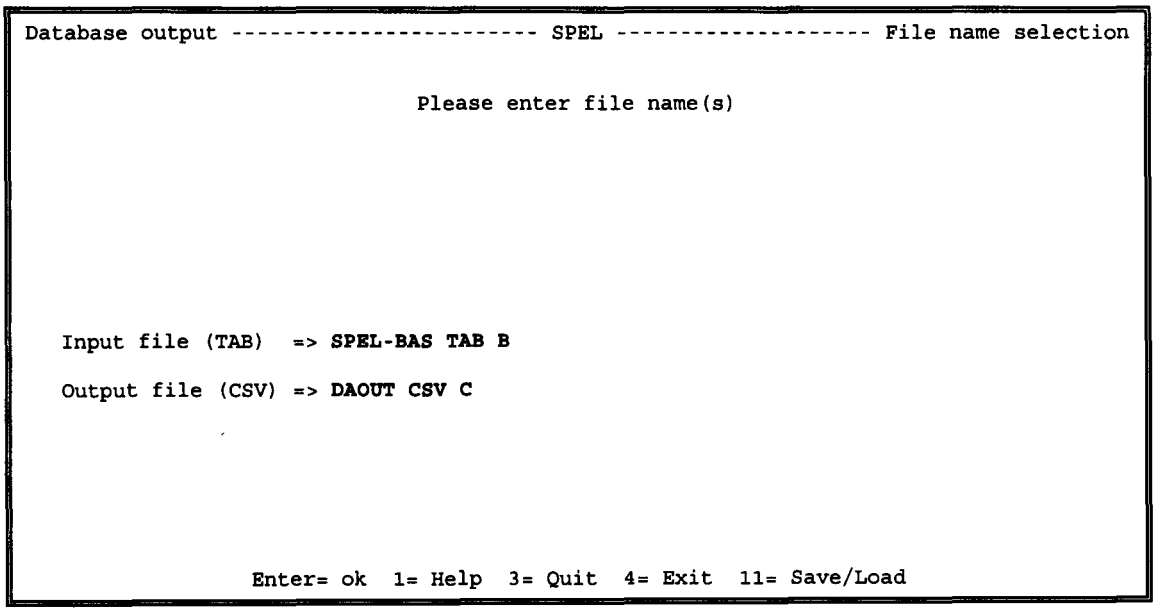
Export data in CSV format

The selected data from the TAB file are stored in a file of CSV format.
This format is used to transfer data to PC standard software, such as EXCEL.

Export data in German CSV format

The same format as CSV, but the ';' is used as column separation.

Work file selection screen



Parameters:

Input file (TAB)

TAB work file containing SPEL tables to print.

Output file (PRN | SDA | CSV | GCS)

Output file of PRN, SDA or CSV format.

Remark: The example above is shown, when 'Export data in CSV format' was selected.

Key selection screen

```
 Database output ---------------------- SPEL ---------------------- Key selection

                     Please select table keys, columns and rows

Region        ( 3 ch.) => F IRL

Sub-region    ( 2 ch.) => 00

Current year  ( 2 ch.) => 78 : 88

Periodicity   ( 2 ch.) => 00

Base year     ( 2 ch.) => NN

Type          ( 4 ch.) => BASB

Model area    ( 1 ch.) => S

Table column  ( 4 ch.) => POTA - RAPE

Table row     ( 4 ch.) => LEVL

                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Region

Region selection. Specify a single region code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*'.

The maximal number of region codes that may be selected is 300.

Sub-region

Sub-regions selection. Specify a single sub-region code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*'.

The maximal number of sub-region codes that may be selected is 10.

Sub-regions are currently not used. The whole region has the sub-region code '00'.

Current year

Year selection. Specify a single year, a sequence of years, a numerical range or a combination of sequences and ranges.

The maximal number of years that may be selected is 50.

Periodicity

Periodicity selection. Specify a single periodicity code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

The maximal number of periodictiy codes that may be selected is 17.

Periodicities are currently not used. The whole year has the code '00'.

Base year

Base year selection. Specify a single base year, a sequence of base years or an alphanumerical range, a combination of sequences and ranges.

The maximal number of base years that may be selected is 10.

For selecting SPEL tables of the ex-post period from the SPEL work file (TAB file) the code for the subkey base year will always be 'NN'.

Type

Type selection. Specify a single type code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*'.

The maximal number of type codes that may be selected is 10.

Model area

Model area selection. Specify a single model area code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*'.

The maximal number of model area codes that may be selected is 5.

Table column

Table column selection. Specify a single table column code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*' .

The maximal number of table column codes that may be selected is 300.

Table row

Table row selection. Specify a single table row code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*' .

The maximal number of table row codes that may be selected is 300.

For detailed information on code selection see chapter 'General program control'.

Remark: The selection must not contain SPEL tables of different table structures.

Transposing control screen

```
Database output ---------------------- SPEL -------------------- Transposing control

                          Please enter transposing modes
    C = Columns, L = Lines, 1 = lowest level tables .... 7 = highest level tables




                Transposing mode for: Region       => 1
                Transposing mode for: Sub-region   => 2
                Transposing mode for: Current year => L
                Transposing mode for: Periodicity  => 4
                Transposing mode for: Base year    => 5
                Transposing mode for: Type         => 6
                Transposing mode for: Model area   => 7
                Transposing mode for: Table column => C
                Transposing mode for: Table row    => 3



                           Enter= ok  1= Help  3= Quit
```

Parameters:

Transposing mode for: ...

Transposing mode control for the displayed dimension. Specify one of the following legal transposing mode codes:

'C' for columns of output tables
'L' for lines of output tables
'1' for output table headers of lowest level
...
'7' for output table headers of highest level.

The decimal codes '1', '2', '3', '4', '5', '6' and '7' control the sequence of the output tables, in which all table headers of level '1' (all regions in the example above) follow each other immediately before the table header of level '2' changes.

The transposing mode codes must be unique.

Output example:

The previous control panel input results in the following printer output:

```
------------------------------------------------
Table     1/001: Region IRL
                 Sub-region 00
                 Table row LEVL
                 Periodicity 00
                 Base year NN
                 Type BASB
                 Model area S
------------------------------------------------
                   1             2             3
                POTA          SUGB          RAPE
------------------------------------------------
 1 78          53.40         35.20          0.00
 2 79          41.20         36.50          0.00
 3 80          40.60         34.90          2.20
 4 81          41.56         32.98          0.00
 5 82          36.00         34.90          0.00
 6 83          37.40         35.50          1.42
 7 84          33.50         36.40          3.03
------------------------------------------------
Date: 28.11.91 14:09:22

------------------------------------------------
Table    1/002:  Region IRL
                 Sub-region 00
                 Table row LEVL
                 Periodicity 00
                 Base year NN
                 Type BASB
                 Model area S
------------------------------------------------
                   1             2             3
                POTA          SUGB          RAPE
------------------------------------------------
 8 85          35.70         34.90          4.45
 9 86          33.00         33.90          4.45
10 87          30.50         37.00          2.02
11 88          30.30         37.10          5.66
------------------------------------------------
Date: 28.11.91 14:09:22
```

6.3. File format conversion

6.3.1. DATCON program: data format conversion

The different original formats of data coming from external originators are converted to the standard IMP format. The format conversion has to be done on a computer system which can access tape units (MVS batch system or VM batch).

The format standardization process, which includes a complete sorting step, is CPU time and I/O consuming. It is therefore recommended that the format conversion should always be run in batch mode.

The format conversion program DATCON is steered by a parameter specifying the source format and a parameter for the type code to be created (FAO data only). The output format is always IMP.

DATCON needs the following parameter block (PAR file) :

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
DATCON
Input format = ........
First year = ..........
END
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
```

Parameters:

Input format

Format of the input file. Legal values are 'FAO' and 'CRONOS'.

First year

Year selection. Specify the first year to take over in the output file.

This parameter block may be generated by a program selection shell. For further information see 'SPEL System, Technical Documentation (Rev. 1), Vol. 2: BS, SFSS, MFSS'.

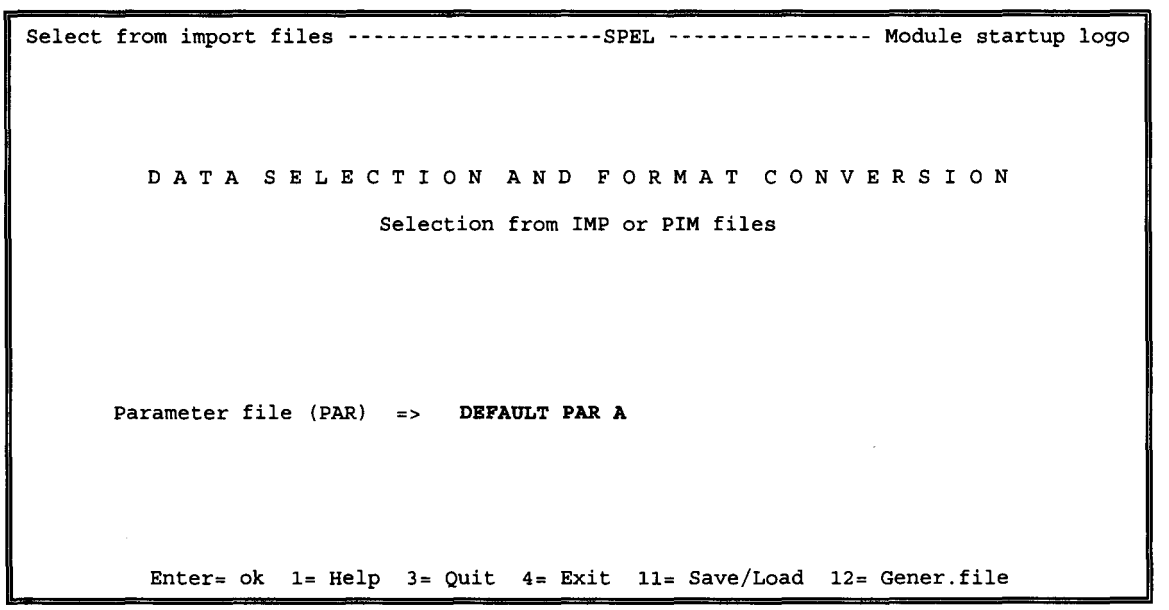
6.3.2. IMPSEL program: data selection on standard import files

Data from IMP or PIM data files can be selected and the format converted by the IMPSEL program. The program works for up to 10 input files of the same format, and builds up one new SDA or IMP output file.

The data selection is done first via a list of selection criteria referring to the identifiers for region, sub-region, year, periodicity and type. Up to five ASS files can be used for a second selection by source code and conversion to SPEL Model codes. If no ASS files are accessed, no further selection is done. For detailed information about ASS files see chapter 'ASS files'.

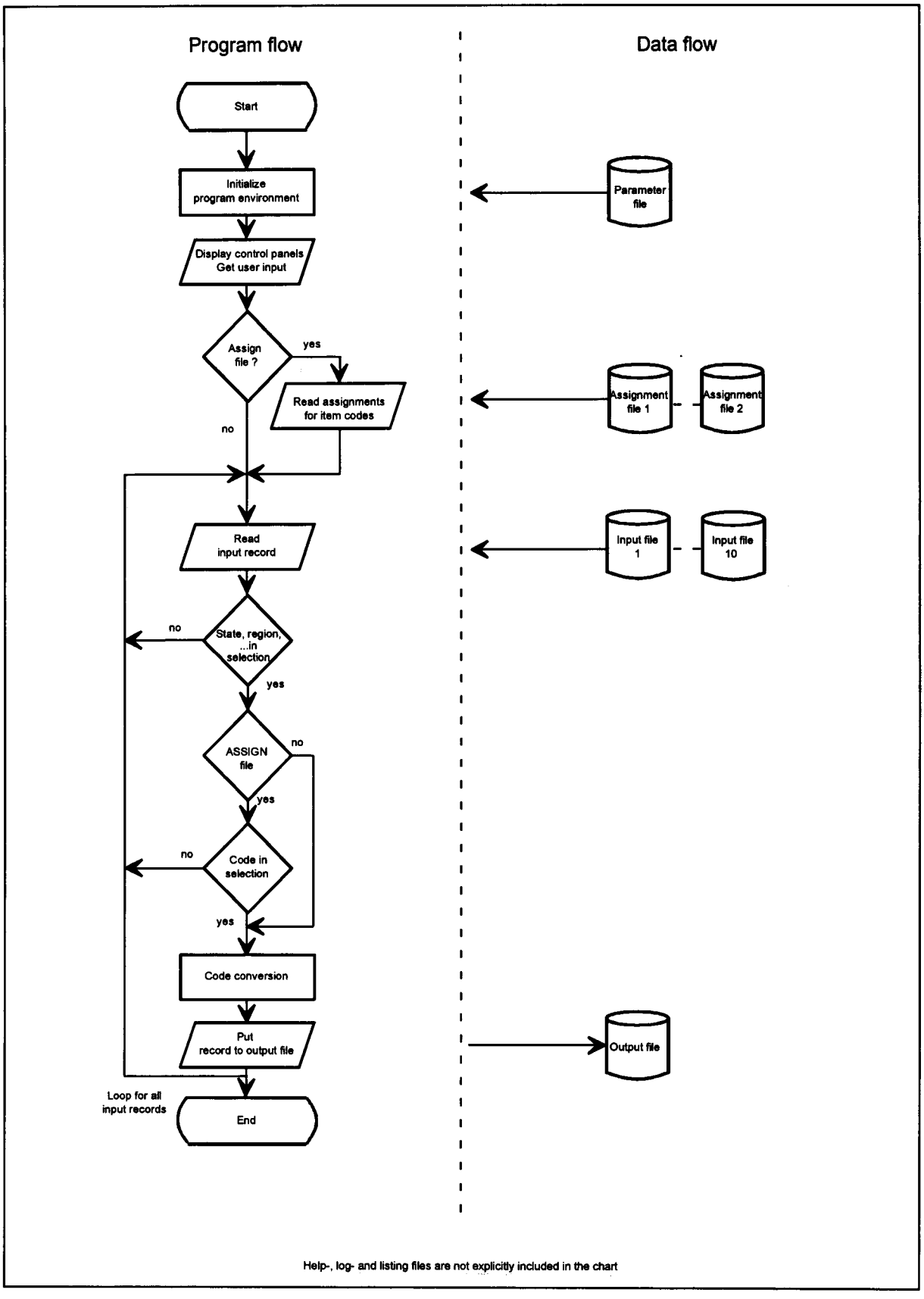
When a format conversion is carried out, it is advisable in most cases to convert the code, because the IMP file codes are source domain codes and those of SDA file types are SPEL Model codes. If no assignment file is accessed, the input file codes will be copied into the output file without conversion.

Module startup screen



For detailed information, see chapter 'General programs'.

Figure 9: IMPSEL program flow chart



Output modus screen

```
Select from import files ------------------ SPEL ---------------------- File formats

                    Please enter file formats for input/output




Input format      (IMP/PIM) => IMP
Output format     (IMP/SDA) => SDA




                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Input format

IMP Files of IMP format
PIM Packed files of PIM format

Output format

For output file format, the user can choose between standard import file format (IMP format) and the standard SDA format.

IMP IMP format is suitable to import source data into the SPEL System. It is optimized for fast item selection.

SDA SDA format is suitable for data to be revised by users using an system editor program.

Working file selection screen

```
 Select from import files ---------------- SPEL ----------------- Work file selection

                                Please enter file names

Input file (IMP): 1   => ZPA1 IMP A
Input file (IMP): 2   =>
Input file (IMP): 3   =>
Input file (IMP): 4   =>
Input file (IMP): 5   =>
Input file (IMP): 6   =>
Input file (IMP): 7   =>
Input file (IMP): 8   =>
Input file (IMP): 9   =>
Input file (IMP):10   =>
Output file (SDA)     => OUTPUT SDA C
Assign file (ASS): 1 => SUPPLY-B ASS F
Assign file (ASS): 2 =>
Assign file (ASS): 3 =>
Assign file (ASS): 4 =>
Assign file (ASS): 5 =>


                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Input file (IMP | PIM): 1 ... 10

The input files have to be of one of the following formats:

- IMP
- PIM

Output file (SDA | IMP)

The output file will be of SDA or IMP format and will be newly formed if you do not enter the name of an existing file.

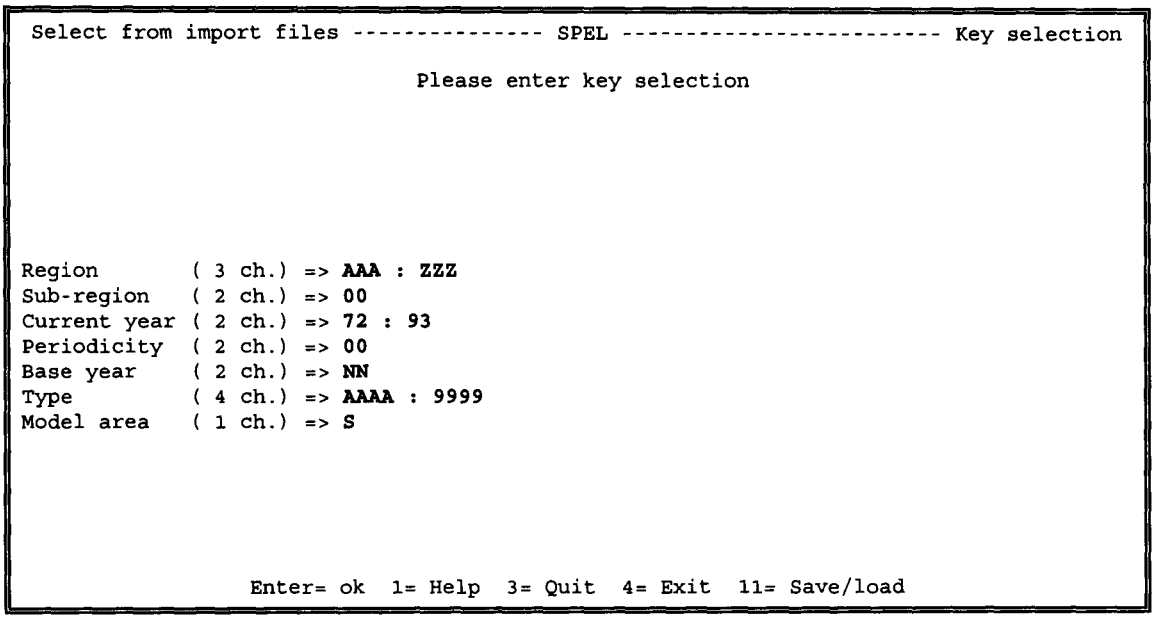
Assign file (ASS):1 ... 5

These ASS files contain assignment statements for selecting source data and converting source data codes to SPEL table element codes (See chapter 'ASS files').

For IMPSEL, the assignment file may be used for additional data selection. The data selection is done first via a list of selection criteria referring to the identifiers for regions, sub-regions, current year, periodicity, base year, type and model area. A second selection step can be added for all defined codes appearing in the assignment statements.

Remark: In the above example the user has selected 'IMP' for input format and 'SDA' for output format.

Selection criteria screen



Parameters:

Region

Region selection. Specify a single region code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

Sub-region

Sub-region selection. Specify a single sub-region code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

Sub-regions are currently not used. The whole region has the sub-region code '00'.

Current year

Year selection. Specify a single year, a sequence of years, a numerical range, a combination of sequences and ranges or '*' .

Periodicity

Periodicity selection. Specify a single periodicity code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

The whole year has the code '00'. Other periodicity codes are currently not used.

Base year

Base year selection. Select one base year, such as '85' or 'NN'.

The specified base year will be used to define the subkey 'base year' in the table keys on the output file of SDA format.

Type

Type selection. Select one type code, such as 'ZPAC' or 'EXPB'

Model area

Model area selection. Select one model area code, such as 'D'.

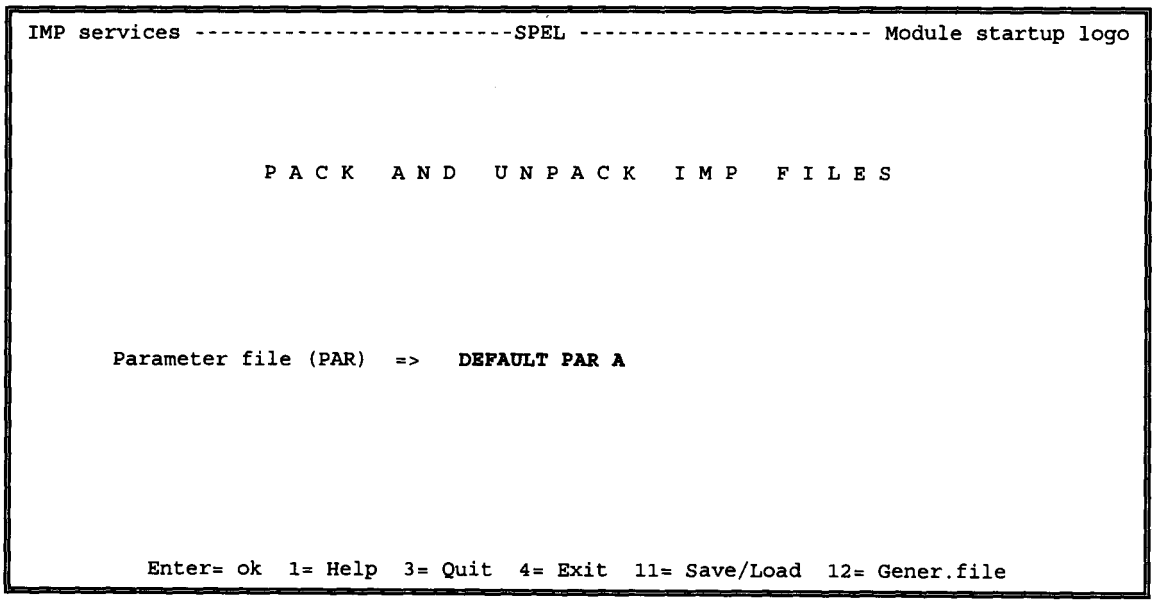
The specified model area will be used to define the subkey 'model area' in the table keys on the output file of SDA format.

For detailed information on code selection see chapter 'General program control'.

6.3.4. IMPPACK program: pack/unpack IMP files

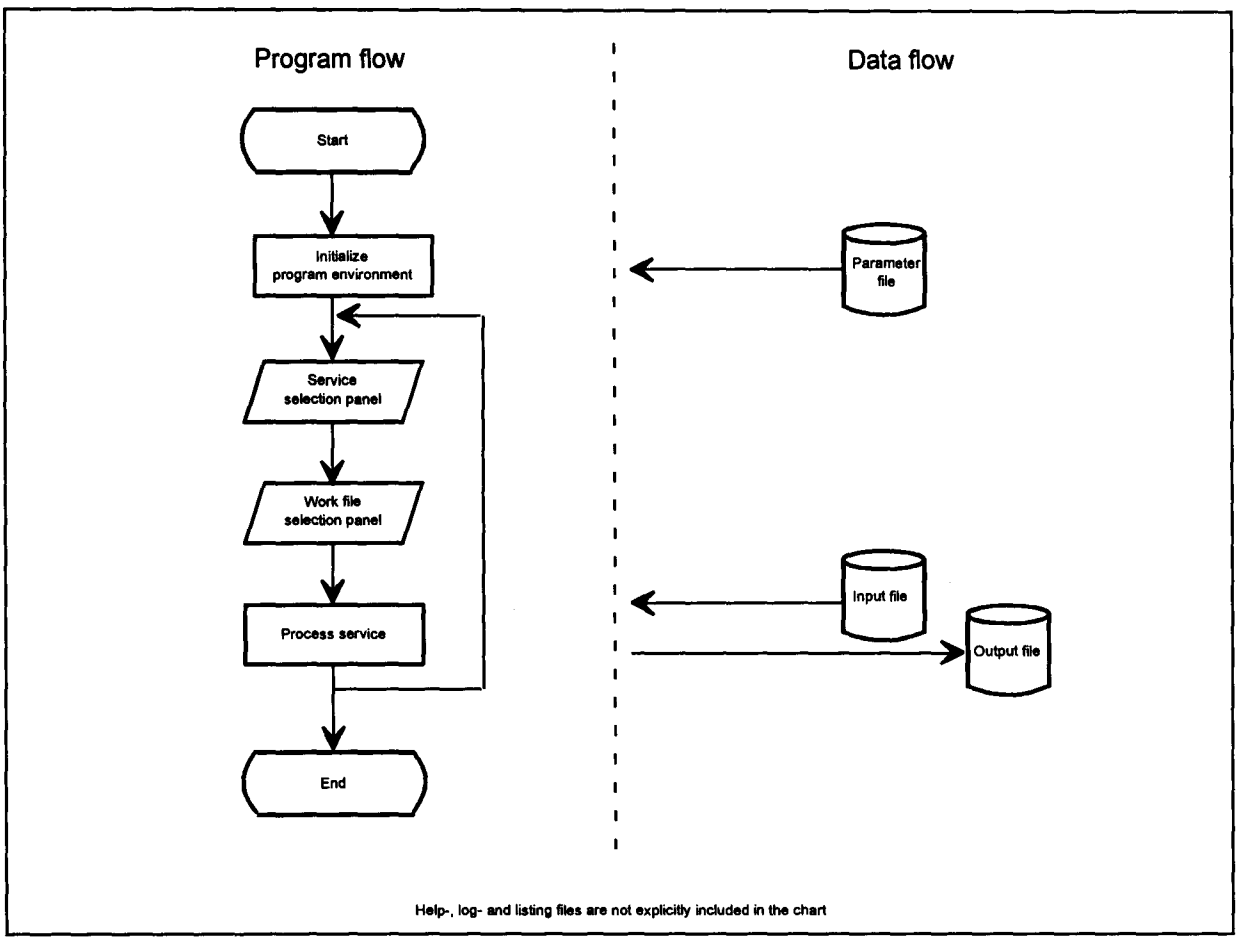
Files of import format IMP can be packed in a compressed form in order to save disk space. These compressed files are of PIM format. The IMPPACK program offers the user the possibility to pack IMP files and unpack PIM files.

Module startup screen



For detailed information, see chapter 'General programs'.

Figure 10: IMPPACK program flow chart



Service selection screen

```
Pack/unpack IMP files -------------------- SPEL ---------- Utility selection

                    Select a service by moving the cursor
                      to any topic below and press ENTER

                          Pack file of IMP format
                          Unpack file of PIM format

                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Selection items:

Pack file of IMP format

Packing a file of IMP format. The output is of PIM format.

Unpack file of IMP format

Unpacking a file of PIM format. The output is of IMP format.

PACK file of IMP format:

Work file selection screen

```
Pack/unpack IMP files-------------------- SPEL ----------------- Work file selection

                              Please select file names




   Input file (IMP)  => ZPA1 IMP a
   Output file (PIM) => ZPA1 PIM a




                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Input file (IMP)

Input file of IMP format to be packed.

Output file (PIM)

Packed output file of PIM format.

UNPACK files of PIM format:

Work file selection screen

```
Pack/unpack IMP files-------------------- SPEL ----------------- Work file selection

                              Please select file names




   Input file (PIM)   => ZPA1 PIM a
   Output file (IMP)  => ZPA1 IMP a




                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Input file (PIM)

Packed input file of PIM format.

Output file (IMP)

Output file of IMP format.

6.4. Trend estimation and data completion

6.4.1. TREND program: trend estimations by single regressions

The TREND program executes trend estimations by single regressions using ordinary least squares (OLS). It is designed to work with simple control for a large quantity of time series.

Input data come from tables stored in a TAB file. Estimation results are again stored in a TAB file.

Default parameters for regression control, the reference period and 'best fit' criteria for automatic selection come from a PAR file revised by panels.

On default for each time series, six OLS estimations are calculated with different data transformations for the dependent variable and the trend. The program automatically selects the results of the

transformation which provides the best fit. Best fit criteria may be the coefficient of determination or Theil's coefficient [7]. The data transformations are:

	dependent	trend
1.	linear	linear
2.	linear	inverse
3.	logarithmic	inverse
4.	logarithmic	logarithmic
5.	linear	logarithmic
6.	logarithmic	linear

Special regression control parameters for specific time series may be selected in the RST control file.

In the SPEL System, TREND is used for filling in missing data in time series as well as for doing trend estimations for simulation years.

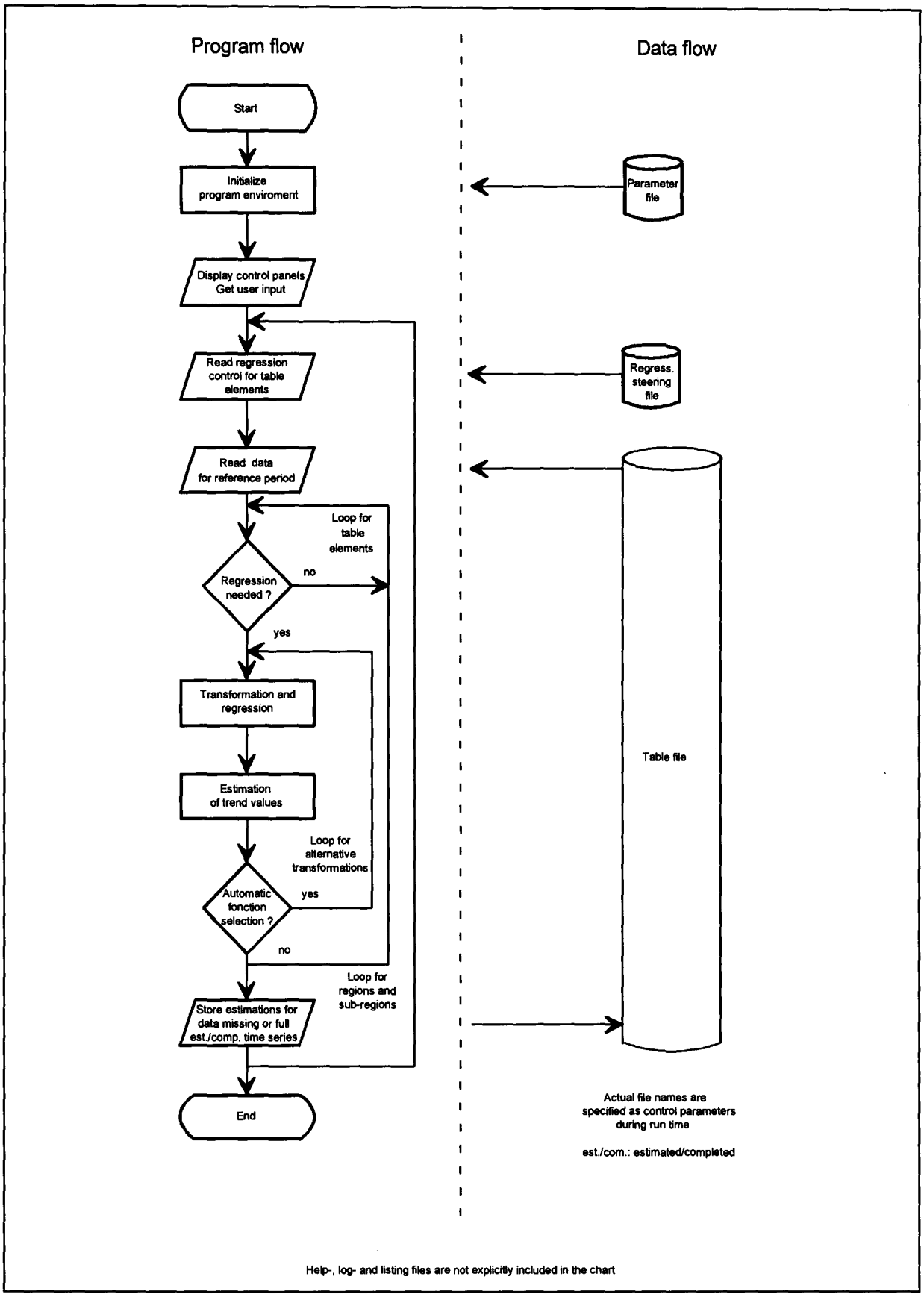
Module startup screen

```
Trend estimation ----------------------- SPEL ------------------ Module startup logo


       T R E N D  E S T I M A T I O N  B Y  S I M P L E  R E G R E S S I O N

                Check and generation of data completeness in time
                                  - or -
                     Trend projection for simulation years.



      Parameter file (PAR)  =>   DEFAULT PAR A


         Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load  12= Gener.file
```

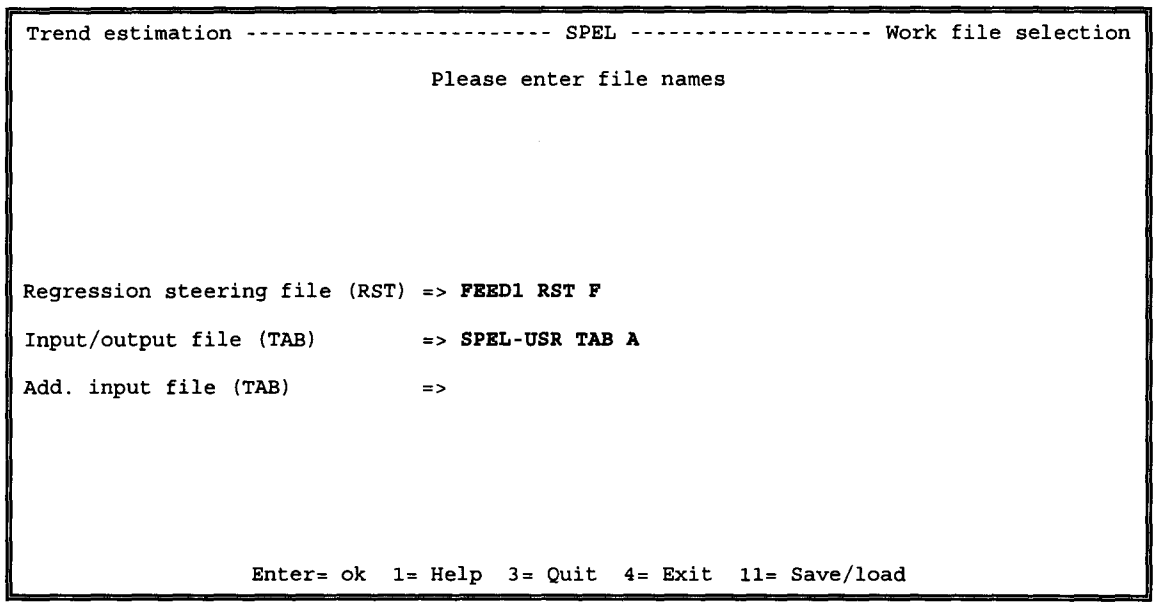
For detailed information, see chapter 'General programs'.

7 For further information see methodological documentation

Figure 11: TREND program flow chart



Work file selection screen



Parameters:

Regression steering file (RST)

This RST file contains explicit definitions for data transformation types and reference intervals for selected time series (See below).

Input/output file (TAB)

This file of TAB format contains SPEL tables.

Add. input file (TAB)

Additional TAB input file only opened for read access. This file is to be accessed for tables not found in the first defined TAB file.

Remark: If only the regression steering file name is specified, TREND will just check a set of trend estimation steering statements for legality. The set to be tested has to be specified by user's selection in the 'Table key selection' panel (next panel).

Table key selection screen

```
Trend estimation ---------------------- SPEL ------------------ Table key selection

                              Please make your selection


Region              ( 3 ch.) => D F

Sub-region          ( 2 ch.) => 00

Output year         ( 2 ch.) => 72 : 89

Periodicity         ( 2 ch.) => 00

Base year           ( 2 ch.) => NN

Table output type   ( 4 ch.) => COMC

Table input types   ( 4 ch.) => ZPAC EXPC ESTC

Model area          ( 1 ch.) => S

                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Region

Region selection. Specify a single region code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*' .

The maximum number of regions that may be selected is 400.

Sub-region

Sub-regions are currently not used.

The whole region has the sub-region code '00'. Always specify '00'.

Output year

Output period selection. Specify a single year, a sequence of years, a numerical range or a combination of sequences and ranges.

The maximum number of years that may be selected is 40.

Periodicity

Periodicity selection. Specify a single periodicity code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*'.

The whole year has the code '00'. Other periodicity codes are currently not used.

When using periodicities the maximum number that may be selected is 19.

Base year

The code for the subkey base year will be always 'NN'.

The subkey cannot be modified.

Table output type

Selection of table type codes for output table.

Specify one legal table type code.

Table input type

Selection of table type codes for input tables. Specify a single type code or a sequence of codes.

The maximum number of input type codes that may be selected is 10.

Model area

Select one model area codes, such as 'S'.

For detailed information on code selection see chapter 'General program control'.

Regression steering screen

```
Trend estimation ------------------------ SPEL ------------------- Steering parameters
                    Please select regression steering parameters



Regression interval            (start : end1 : end2)  =>  72 : 85 : 89
Significance coefficient        (THEIL/DETERMINATION) => DETERMINATION
Output level     (NONE/MISSINGS/FULL TREND/COMPLETED) => COMPLETED
General list level          (FULL/ROUGH/MINIMUM/NONE) => NONE
Special list level steering                  (YES/NO) => NO



                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Regression interval

Define the years for short and long regression reference intervals.

syntax:

start : end1 : end2

start : end1 will be the short interval
start : end2 will be the long interval

end1 and *end2* may be the same year .

For the short interval you must specify at least 4 years and the start year has to precede the end years.

Example: 72 : 80 : 89 or 80 : 90 : 90

Significance coefficient

Select one of the following for best fit criteria:

THEIL	Theil's coefficient[8]
DETERMINATION	coefficient of determination[9].

Output level

For output into the TAB work file you can choose between four levels:

NONE	nothing is stored
MISSINGS	only the estimated values for missing data are stored
FULL TREND	full trend time series are stored.
COMPLETED	time series including original values and estimates of missing data are stored.

General list level

You can choose between the following list levels:

'FULL' provides, for each time series, listing on
- steering parameters
- original values for the whole of reference and output period
- results of all data transformation alternatives
- the selected transformation type
- the completed time series.

---

8 $\text{THEIL} = \frac{\text{Sum of squares of deviation}}{\text{Sum of squares of yearly changes}}$

Source: Henri Theil, Applied Economic Forecasting, Amsterdam 1971, page 28

9 $\text{DETERMINATION} = \frac{\text{Sum of squares explained}}{\text{Sum of squares total}}$

'ROUGH' provides, for each time series, listing on
- steering parameters
- original values for the whole of reference and output periods
- the completed time series.

'MINIMUM' provides, for each time series, listing on
- original values for the whole of reference and output periods
- the completed time series.

'NONE' nothing will be listed.

Special list level steering

Select YES if you want time series selected to be listed in a different level as the selected general level, otherwise select NO.

List level steering screen

```
Trend estimation ----------------------- SPEL ------------------ List level steering

              Please select groups and list levels
                        example: SC,PROP,*,ROUGH
                                 |    |  |    | List level
                                 |    |  | rows
                                 |    | columns
                                 | structure: model area + table structure type



Listing group:  1 =>SC,PROP,SWHE - OCER,MINIMUM
Listing group:  2 =>SC,PROP,BARL,FULL
Listing group:  3 =>
Listing group:  4 =>
Listing group:  5 =>
Listing group:  6 =>
Listing group:  7 =>
Listing group:  8 =>
Listing group:  9 =>
Listing group: 10 =>


                   Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Listing group

For groups of columns and row codes which identify the time series of the SPEL database the user can define special list levels.

In a first step the list level for all time series is set to the general list level as defined in the 'Regression steering panel'. Interpreting the input of the 'List level steering panel', special list levels may be set for one or a group of time series. A listing group with a higher label redefines the ones with the lower labels, if the higher label group is contained in the lower label groups (e.g. if listing group n 'wheat' is also contained in listing group n-1 'cereals').

syntax:

ms,columnCode,rowCode,listlevel

ms combination of model area and table structure type
m specify one model area code such as 'S'
s specify 'C' for complementary table or 'B' for basic table

columnCode Specify a single column code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*'.

rowCode Specify a single row code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*'.

listlevel Choice between the following list levels:
FULL | ROUGH | MINIMUM | NONE
For detailed information, see previous 'Regressing steering panel', parameter 'General list level'.

6.4.1.1. RST files

RST files are used to define steering parameter values for selected trend estimations which are different from the general values. These general values are defined in the sequence of the screen panels when calling the program. The RST files are sequential files with a fixed record length of 72.

File structure:

There are three record types:

- Comment records

Beginning with an '*' at the first character position. Comments may occur anywhere in the file.

- Definition block records.

This block is optional, but if used it must appear on top of the file. It specifies how to handle SPEL table elements not explicitly defined in the file's regression steering records. The block consists of three records with identifiers beginning in the first column:

First record DEFINE

Second record INCLUDE or EXCLUDE

INCLUDE All SPEL table elements are implicitly included in the regression and trend estimation.
EXCLUDE All SPEL table elements are implicitly excluded from the regression and trend estimation.

Third record END

If no definition block exists, INCLUDE is assumed.

- Regression steering records

All records have the following unique structure :

sss xx ccccrrrr [dddiii ff:mm:ll]

where

pos. 01		must be blank
pos. 02-04	*sss*	Region code
pos. 05		*Blank*
pos. 06-07	*xx*	Sub-region code
pos. 08		*Blank*
pos. 09-16	*ccccrrrr*	Table element code (col/row)

For all these positions, wildcard characters such as '*' may be used.

pos. 18-23	*dddiii*	Data transformation type	
		ddd	dependent variable (aligned left)
		iii	independent variable (aligned left)
		Blank	means automatic selection by best fit with the following data transformation types: LINLIN LININV LN INV LN LN LINLN LN LIN
pos. 24		*Blank*	
pos. 25-32	*ff:mm:ll*	*ff,mm* and *ll* are limits of the reference interval The interpretation is application-dependent.	
pos. 33		*Blank*	
pos. 34-72		Reserved for special extensions, e.g. applications in work sector SFSS of the SPEL/EU-Model. All extensions are decribed with the documentation of these applications.	

The data transformation types for *ddd* and *iii* are as follows:

LIN	Linear (no transformation)	$x' = x$
LN	Natural logarithm	$x' = \mathrm{LOG}_e(x)$
LOG	Logarithm base 10	$x' = \mathrm{LOG}_{10}(x)$
SQR	Square root	$x' = \sqrt{x}$
REZ	Inverse	$x' = \frac{1}{X}$
EXP	Exponential	$x' = e^X$
POW	Power	$x' = 10^X$
QUA	Square	$x' = x^2$

Example of a RST file

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
*
*  define default modus
DEFINE
INCLUDE
END
*  regression steering definitions
*  (only format examples)
 D    00 LEVLSWHE LOGLIN 75:80:87
 F    00 LEVLSWHE LN LIN 76:80:87
 NL   00 *    *     LINLIN
 E    00 *    *           80:85:90
*
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
```

6.4.2. COMPLET program: fill gaps in time series

The COMPLET program fills gaps in time series of the SPEL database, taking into account the methodological structure of the SPEL table.

Input data come from tables stored in a TAB file. Estimation results are again stored in a TAB file.

The COMPLET program is steered in combinations of panels and the CST steering language. The panel inputs define the overall available elements of each dimension (regions, years etc.), whereas the steering language allows the user to select tables by region, type and model area and to define individual calculations of time series identified by a combination of column and row codes.

The set of possible calculations of the COMPLET program fall into two distinct groups:

- Estimations to fill gaps in time series

  For this purpose, two different methods, both using dependencies of time series, are offered:

  The 'RATIO' method calculates 'data missings' within the dependent time series by applying the relative change of independent variable to dependent variable using the average of the first *n* not missing observations. If the number *n* is not defined by the user, n=3 is assumed.

  The 'REGRESS' method calculates 'data missings' within the dependent time series by single equation OLS estimations, where the user can define the data transformation type for the independent and for the dependent variable. If the transformation types are not user-defined, regressions are calculated for six fixed data transformations and an automatic selection by coefficient of determination is made. For trend estimations, the independent time series is the vector of selected years. The following table shows the fixed data transformations.

	dependent	independent
1.	linear	linear
2.	linear	inverse
3.	logarithmic	inverse
4.	logarithmic	logarithmic
5.	linear	logarithmic
6.	logarithmic	linear

For both methods, 'RATIO' and 'REGRESS', the independent and the dependent time series are checked before calculation. The independent must not include 'data missings'. If there are no data for the dependent, it may be replaced by the independent, provided a special option for this dependent variable is set in the steering language.

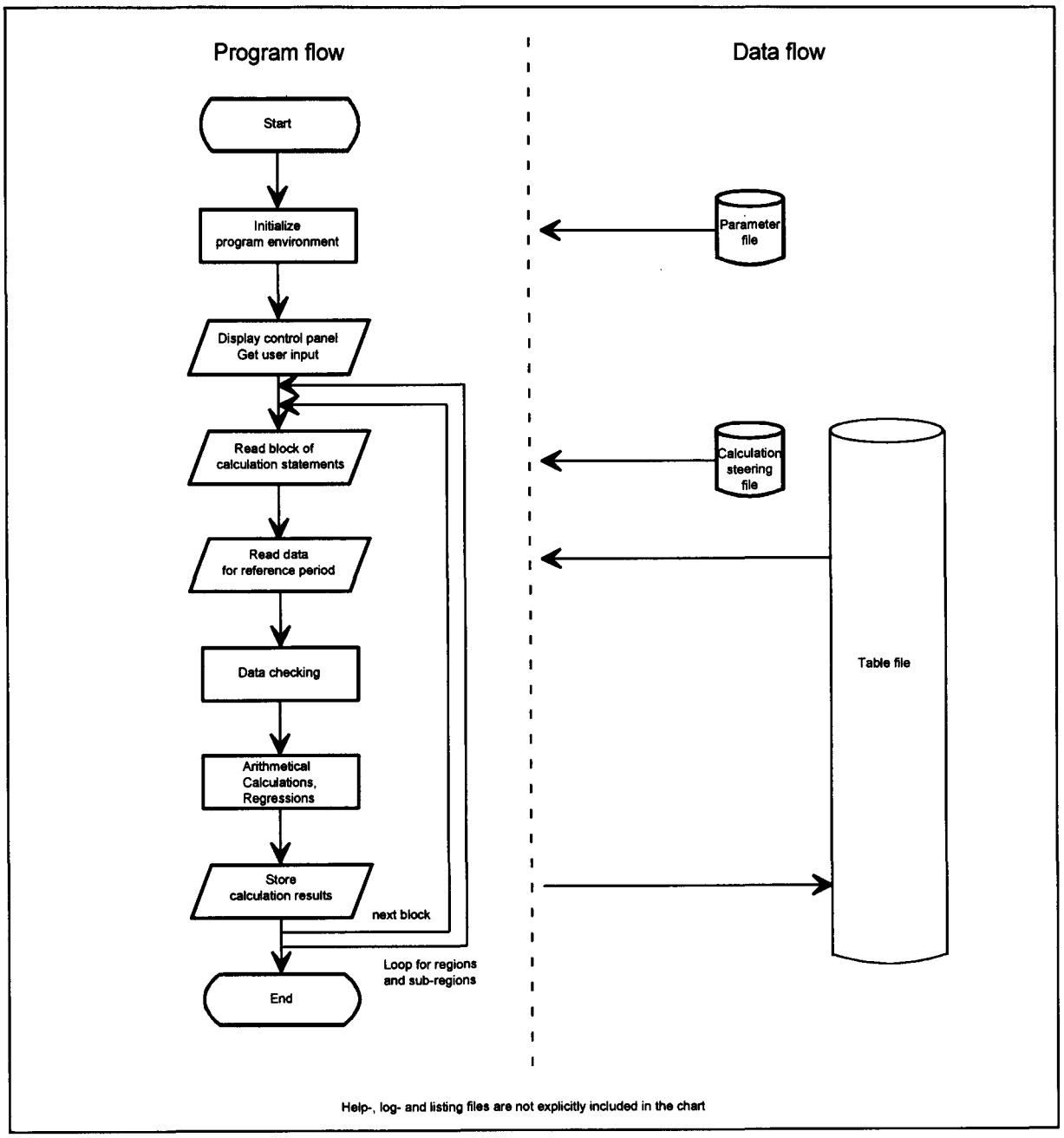
Calculations are always made for two intervals. The end of the first interval has to be defined in a panel user interface. This procedure ensures that the data behaviour of a specific interval only affects the estimations of this interval. If the user wants to work only with one interval, the end of the first interval has to be set to the end of the calculation period.

- Arithmetical operation on time series

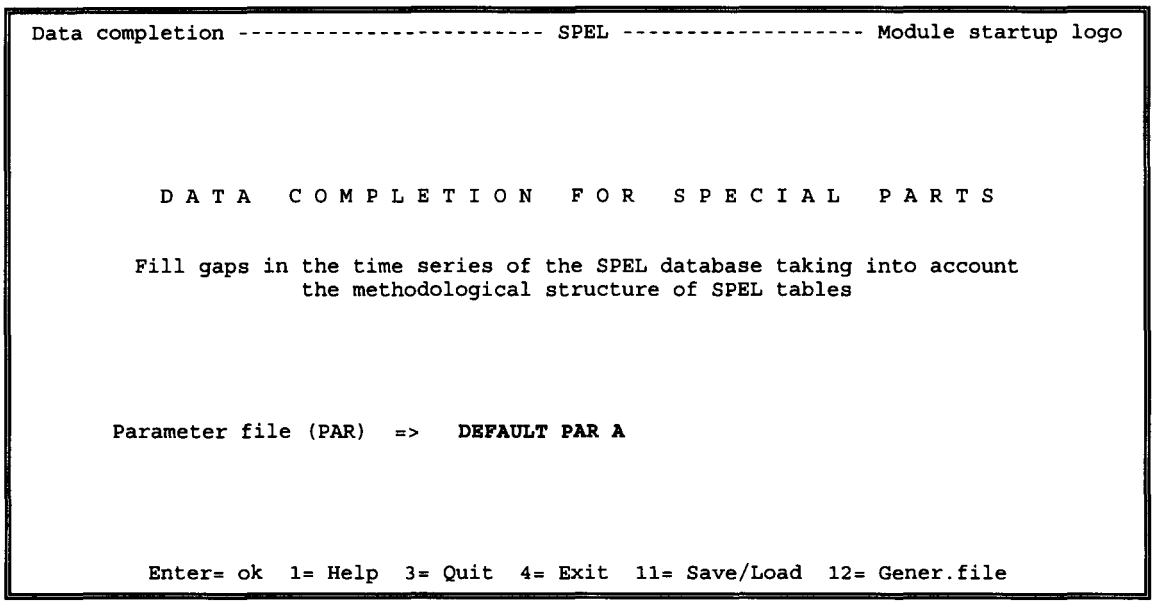
  A formula interpreter is implemented in the COMPLET program. Within the steering language, the user can define arithmetical operations on time series.

For detailed information on the CST steering language, see below chapter 'CST files.

Figure 12: COMPLET program flow chart

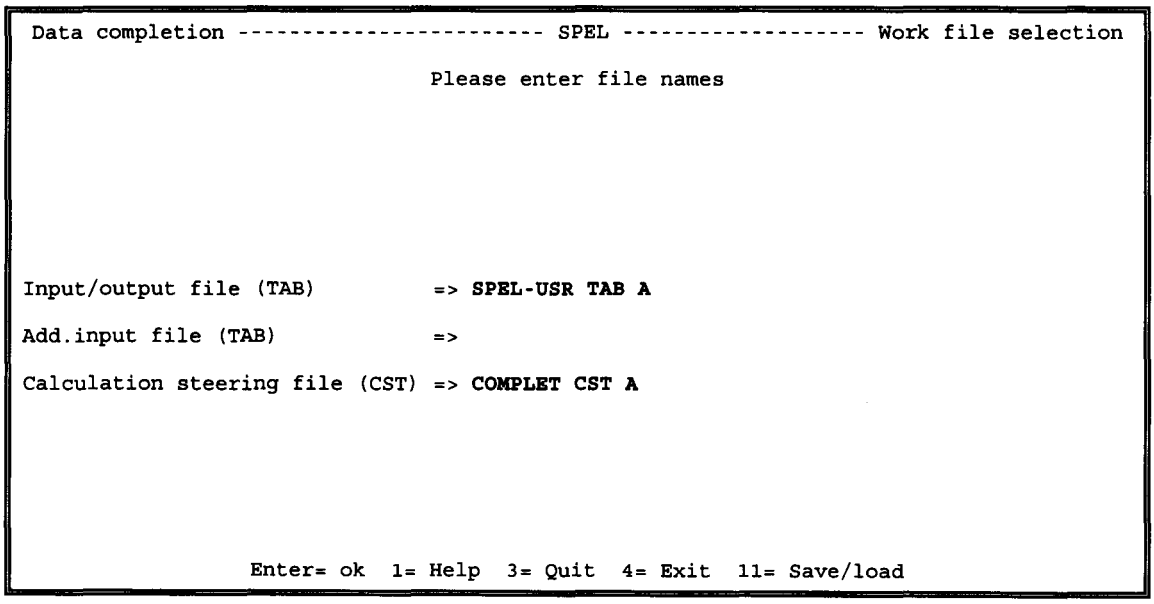


Module startup screen



For detailed information, see chapter 'General programs'.

Work file selection screen



Parameters:

Input/output file (TAB)

This TAB file contains SPEL tables.

Add. input file (TAB)

Additional TAB input file. This file is to be accessed for tables not found in the first defined file.

Calculation steering file (CST)

This CST steering file includes the calculation statements.

For detailed information about CST files, see below chapter 'CST files.

Table key selection screen

```
Data completion ---------------------- SPEL ------------------ Table key selection

                            Please make your selection



Region               ( 3 ch.) => D F
Sub-region           ( 2 ch.) => 00
Current year         ( 2 ch.) => 72 : 93
Periodicity          ( 2 ch.) => 00
Base year            ( 2 ch.) => NN
Table output type    ( 4 ch.) => CMPC
Input types (indep.) ( 4 ch.) => ZPAC COSC PRAC SECC EXPC ESTC CMPC COMC
Input types (dep.)   ( 4 ch.) => ZPAC COSC PRAC SECC EXPC ESTC CMPC
Model area           ( 1 ch.) => S



                Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Region

Region selection. Specify a single region code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*' .

The maximum number of regions that may be selected is 20.

Sub-region

Sub-regions are currently not used.

The whole region has the sub-region code '00'. Always specify '00'.

Current year

Year selection. Specify a single year, a sequence of years, a numerical range or a combination of sequences and ranges.

The maximum number of years that may be selected is 30.

Periodicity

Periodicities are currently not used.

The whole year has the periodicity code '00'. Always specify '00'.

Base year

Select one base year, such as 'NN' or '91'

Table output type

Selection of a table type code for output tables.

Specify one legal table type code.

Input types (indep.)

Selection of table type codes for input tables of the independent time series. Specify a single type code or a sequence of codes . Time series are filled by the first existing data that are found in the sequence of type codes from left to right.

The maximum number of input type codes that may be selected is 12.

Input types (dep.)

Selection of table type codes for input tables of the dependent time series. Specify a single type code or a sequence of codes .. Time series are filled by the first existing data that are found in the sequence of type codes from left to right.

The maximum number of input type codes that may be selected is 12.

Model area

Select one model area codes, such as 'S'

For detailed information on code selection see chapter 'General program control'.

Steering parameter screen

```
Data completion ---------------------- SPEL ------------------ Steering parameters

                      Please select interval and output levels




Last year of first interval          (e.g. 85 =1985) => 89

Output level               (MISSINGS/COMPLETED/NONE) => MISSINGS

Reset output tables                          (YES/NO) => YES

General list level                 (FULL/ROUGH/NONE) => NONE

Special list level steering                  (YES/NO) => NO



                 Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Last year of first interval

Define a code, such as '89' for the year 1989:

This year will indicate the end of the first of two intervals for the calculation methods 'RATIO' and 'REGRESS', (See below chapter 'CST files').

Output level

For output into the TAB file, the user can choose between three levels:

NONE	nothing is stored.
MISSINGS	only the estimated values for missing data are stored.
COMPLETED	time series including original values and estimates of missing data are stored.

Reset output tables

YES	the output tables are reset to data missing at program start.
NO	the output tables will be updated by new calculation results.

General list level

You can choose between the following list levels:

'FULL' provides, for each calculation, listings as follows:
- for 'RATIO' and 'REGRESS': data of the independent and dependent time series
- for arithmetical operations: data of the operands
- for 'REGRESS': results of all data transformation alternatives and the selected transformation type
- for 'RATIO': the relative change of the independent variable
- the resulting time series.

'ROUGH' provides, for each calculation, listings as follows:
- for 'RATIO' and 'REGRESS': data of the independent and dependent time series
- for arithmetical operations: data of the operands
- the resulting time series.

'NONE' nothing will be listed.

Special list level steering

Select YES if you want to select time series to be listed at a special level.
Select NO to use always the general list level.

List level steering screen

```
Data completion ---------------------- SPEL ------------------ List level steering

             Please select groups and list levels
                       example: SC,PROP,*,ROUGH
                                |   |  |    | List level
                                |   |  | rows
                                |   | columns
                                | structure: model area + table structure type



Listing group:  1 =>SC,PROP,SWHE - OCER,MINIMUM
Listing group:  2 =>SC,PROP,BARL,FULL
Listing group:  3 =>
Listing group:  4 =>
Listing group:  5 =>
Listing group:  6 =>
Listing group:  7 =>
Listing group:  8 =>
Listing group:  9 =>
Listing group: 10 =>


                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/load
```

Parameters:

Listing group

For groups of columns and row codes which identify the time series of the SPEL database, the user can define special list levels.

In a first step, the list level for all time series is set to the general list level as defined in the 'Steering parameter panel'. When interpreting the input of the 'List level steering panel', special list levels may be set for one or a group of time series. A listing group with a higher label will redefine the list level of time series also contained in ones with lower level.(e.g. if listing group n 'wheat' is also contained in listing group n-1 'cereals').

syntax:

ms,columnCode,rowCode,listlevel

ms	combination of model area and table structure type *m* specify one model area code such as 'S' s specify 'C' for complementary table or 'B' for basic table
columnCode	Specify a single column code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*'.
rowCode	Specify a single row code, a sequence of codes, an alphanumerical range, a logical range, a combination of sequences and ranges or '*',
listlevel	Choose between the following list levels: FULL | ROUGH | NONE For detailed information, see previous 'Steering parameter panel', parameter 'General list level'.

6.4.2.1. CST files

The panel inputs of the COMPLET program define the available elements of each dimension (regions, years etc.) from which the steering language in the CST files allows a selection of tables according to region, type and model area. The language also defines individual calculations of time series according to a combination of column and row codes. Additional column and row codes for interim results may be defined using the character '&' in the first code position. For trend estimations, the independent variable is identified by the reserved code '&TRD&TRD' and is generated by the selected years, e.g. the first selected year is 1973 then the first element will be 73.

The CST files are sequential files with a fixed record length of 255. For user friendly editing, the line length should not exceed 72 characters.

Language structure:

Calculations made according to a specific selection of regions, types and model area are collected in a block.. Each block begins with the definition statement SELECT and ends before the next SELECT or end of file. Within these blocks, the calculations are defined by calculation statements and the results can be stored in a TAB file using the PUT statement. Each statement may be continued over several file records.

- Comment statement

Record comments begin with an '*' in the first character position. Comments may appear anywhere in the file.

- SELECT statement

General steering parameters are set by panels within the screen panel sequence of the COMPLET program. The SELECT statements permit the option to redefine these parameters ready for the next calculation block (e.g. to select some other regions from the panel selection for the next calculations).

Statement syntax:

SELECT [REGION=*region,*] [MODELAREA=*moda,*] [TYPINDEP=*itypes,*]
[TYPDEP=*dtypes,*] [COMMENT=*text*];

Each statement must begin with the keyword SELECT and must end with the delimeter ';'. The parameter values may be enclosed by quotation marks.

Parameter REGION:

region Sequence and/or range of character region codes (e.g.: D F E).

Parameter MODELAREA:

moda One model area code such as 'S'.

Parameter TYPINDEP:

itype Sequence of table type codes which are used for completing data in the time series of all independent variables defined in the following calculation statements. Time series are filled by the first existing data that are found in the sequence of the type codes from left to right.
(e.g. TYPINDEP = ZPAC ESTC COMC)

Parameter TYPEDEP:

dtype Sequence of table type codes which are used for completing data in the time series of all dependent variables defined in the following calculation statements. Time series are filled by the first existing data that are found in the sequence of the type codes from left to right.
(e.g. TYPDEP = ZPAC ESTC)

Parameter COMMENT:

text Comment text to be printed in the LOG file.

- ESTIMATE statement

Each ESTIMATE statement defines calculations to complete gaps in missing data from specific dependent time series, specified by lists of column and row codes. The calculations are always done for two intervals of time, set in a panel of the COMPLET program.

Statement syntax:

ESTIMATE COLUMNS=*columns*, ROWS=*rows*,
METHOD = RATIO *y* | REGRESS *ttc ttr*,
[MISSDEP = IGNORE | REPLACE,] [MISSINDEP = DMISS | ZERO,]
[MODE = LINKED,] [LAG = *lag,*] [COMMENT = *text,*];

Each steering statement must begin with the keyword ESTIMATE and must end with the delimeter ';'. The parameter values may be enclosed by quotes.

Parameter COLUMNS:

columns Definition of dependency for columns.

Syntax of the parameter value:
indep 1 indep 2..... [< *dep 1 dep 2 dep 3 .....*]
$indep_i$ column code of independent variable i.
dep_i column code of dependent variable i.

If no codes for the dependent variable are given, the codes of the independent variable are implicitly used. Logical ranges defined by column codes such as PLOF - PCOF, are expanded according to the definitions in the COD file.
(See the example below)

Parameter ROWS:

rows Definition of dependency for rows.

Syntax of the parameter value:
indep 1 indep 2 .....[< *dep 1 dep 2 dep 3 .....*]
$indep_i$ row code of independent variable i.
dep_i row code of dependent variable i.

If no codes for the dependent variable are given, the codes of the independent variable are implicitly used. Logical ranges defined by row codes such as SWHE - OCER, are expanded according to the definitions in the COD file.
(See the example below)

Parameter METHOD:

RATIO *y* Missing data within the dependent time series should be calculated applying the relative change of the independent variable to dependent variable, using the average of the first *y* available observation in each interval as a basis.
If *y* is missing y=3 is assumed.

REGRESS *ttc ttr* Missing data within the dependent time series should be calculated by single equation OLS estimation, where *ttc* is the data transformation type for the independent and *ttr* for the dependent variable.

Legal transformation types are:
LIN, LN, LOG, SQR, REZ, EXP, POW and QUA (See chapter 'RST files').
If the transformation types are missing, an automatic selection by coefficient of determination is made with the following data transformation types:
LIN LIN, LIN INV, LN INV, LN LN, LIN LN, LN LIN

Parameter MISSDEP:

IGNORE No estimates are made for dependent time series that include no data. MISSDEP=IGNORE is the default value.

REPLACE If there are no data for the dependent time series it is replaced by the independent time series.

Parameter MISSINDEP:

DMISS If there is no data for the independent time series no replacing will occur.
MISSINDEP=DMISS is the default value.

ZERO If there is no data for the independent time series, it is set to ZERO (= 0.0).

Parameter MODE:

LINKED Missing data in the independent time series are filled by preceding calculation results.

Parameter LAG:

lag Time lag for dependent variable in years
(e.g. LAG = 1 means data for year i area shifted to year i+1).

Parameter COMMENT:

text Comment text to be printed in the LOG file.

Example:

```
ESTIMATE  COLUMNS = HLEV < LEVL, ROWS='CERT < SWHE - MAIZ',
          METHOD=RATIO,MODE=LINKED;
```

1. The independent column code is HLEV and the dependent code is LEVL
2. The independent row code is CERT. The dependent row codes range between SWHE and MAIZ, according to the row code sequence of the SPEL table.
3. The estimation method is RATIO, which uses 3 year average.
4. Results of previous calculations are used for the independent time series.
5. The calculation loops will be :
   estimate for both intervals as defined by the interval parameter HLEVSWHE by HLEVCERT until HLEVMAIZ by HLEVCERT
   estimate for both intervals as defined by the interval parameter LEVLSWHE by HLEVSWHE until LEVLMAIZ by HLEVMAIZ.

- ARITH statement

The arithmetic statements define arithmetical instructions on time series.

Statement syntax:

ARITH *resultCode* = *arithmeticalExpression*, [COMMENT = *text*] ;

Each steering statement must begin with ARITH and must end with the delimeter ';'.

resultCode String of up to 11 characters, divided into 3 parts, where each part has to be separated by the character '.'

part 1: fixed identifier 'E' to identify this code as a resulting time series
part 2: a SPEL table column code of up to 4 characters
part 3: a SPEL table row code of up to 4 characters

(See the example below)

arithmeticalExpression string including the arithmetical instruction, where the operandsmay be numerical constants or codes of time series.
Legal operators are:
'+', '-', '*', '/', '**', SQRT, LN, LOG, SIN, COS, TAN.
Brackets are allowed.

The syntax of the codes for the operands is the same as for the result code, except that for part 1 the following identifiers can be used:
'E' to use an estimated time series
'D' to use a dependent time series
'I' to use an independent time series
(See SELECT statement, parameters TYPDEP and TYPINDEP)

(See the example below)

Parameter COMMENT:

text Comment text to be printed in the LOG file.

Example:

```
ARITH  E.PROP.SWHE = D.SWHE.SWHE  *  D.SWHE.LEVL  /  1000,
COMMENT='Calculation of soft wheat production from yield coefficient and area' ;
```

- PUT statement

The PUT statements are for output control. When doing the calculation the results are collected in SPEL tables. Using the put statement, the selected columns and rows are stored in a file of TAB format.

Statement syntax:

PUT COLUMNS = *columns*, ROWS = *rows*.;

Each steering statement must begin with the keyword PUT and must end with the delimeter ';'. The parameter values may be enclosed by quotes.

Parameter COLUMNS:

columns Sequence and/or range of character codes for table columns to store (e.g. COLUMNS = 'SWHE - OCER')

Parameter ROWS:

rows Sequence and/or range of character codes for table rows to store (e.g. ROWS = 'PROP LEVL GVAM')

Example of a CST file

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
* ********************************************************************
*                        crop production data
* ********************************************************************
SELECT    COMMENT='Trend for yield coefficient', MODELAREA='S',
          TYPDEP='BASB',TYPINDEP='BASB';
*
ARITH     E.&CRT.LEVL = D.SWHE.LEVL+D.DWHE.LEVL+D.BARL.LEVL+D.OCER.LEVL+
                        D.OATS.LEVL+D.RYE.LEVL+D.MAIZ.LEVL;
ARITH     E.SWHE.LEVL = D.SWHE.LEVL/E.&CRT.LEVL;
ARITH     E.DWHE.LEVL = D.DWHE.LEVL/E.&CRT.LEVL;
ARITH     E.BARL.LEVL = D.BARL.LEVL/E.&CRT.LEVL;
ARITH     E.OCER.LEVL = D.OCER.LEVL/E.&CRT.LEVL;
ARITH     E.OATS.LEVL = D.OATS.LEVL/E.&CRT.LEVL;
ARITH     E.RYE.LEVL  = D.RYE.LEVL /E.&CRT.LEVL;
ARITH     E.MAIZ.LEVL = D.MAIZ.LEVL/E.&CRT.LEVL;
*
 ESTIMATE COLUMNS='&TRD < &CRT',ROWS='&TRD < LEVL',MODE=LINKED,
          METHOD=REGRESS;
 ESTIMATE COLUMNS='&TRD < SWHE - OCER',ROWS='&TRD < LEVL',MODE=LINKED,
          METHOD=REGRESS;
*
ARITH     E.&CRS.LEVL = E.SWHE.LEVL+E.DWHE.LEVL+E.BARL.LEVL+E.OCER.LEVL+
                        E.OATS.LEVL+E.RYE.LEVL+E.MAIZ.LEVL;
ARITH     E.SWHE.LEVL = E.SWHE.LEVL  / E.&CRS.LEVL * E.&CRT.LEVL;
ARITH     E.DWHE.LEVL = E.DWHE.LEVL  / E.&CRS.LEVL * E.&CRT.LEVL;
ARITH     E.BARL.LEVL = E.BARL.LEVL  / E.&CRS.LEVL * E.&CRT.LEVL;
ARITH     E.OCER.LEVL = E.OCER.LEVL  / E.&CRS.LEVL * E.&CRT.LEVL;
ARITH     E.OATS.LEVL = E.OATS.LEVL  / E.&CRS.LEVL * E.&CRT.LEVL;
ARITH     E.RYE.LEVL  = E.RYE.LEVL   / E.&CRS.LEVL * E.&CRT.LEVL;
ARITH     E.MAIZ.LEVL = E.MAIZ.LEVL  / E.&CRS.LEVL * E.&CRT.LEVL;
*
PUT       COLUMNS=*, ROWS=*;
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
```

6.5. Evaluation

6.5.1. EV program: general estimation and evaluation program

6.5.1.1. Application abstracts

The EV program offers general evaluations and single equation regression estimations based on the standard file formats TAB and SDA supported by the SPEL System.

EV has its own command language for both batch and interactive operations. In interactive mode the commands are entered directly from the keyboard and executed immediately. For batch operations the commands are collected in a STE control file.

The simplest use of the EV program is to obtain data, e.g. the production of wheat in a specific period for specified sequences of regions (e.g. Member States). Such data are provided in the form of tables for printing or listing on the screen. Growth rates and averages, etc. can be calculated.

The program also permits the analysis of data in the econometrical sense based on single equation regressions with one or more independent variables. Quality indicators of such regression estimates can be accessed on the basis of the coefficient of determination, correlation coefficient, T-value, F-value, standard of estimation error, Durbin-Watson coefficient, Theil's coefficient, etc.

In combination with large sets of commands collected in STE files it is also used to carry out certain work steps of the SPEL System.

Most of the EV operations are defined at the 'EV-tables' level.

These tables are not the same as SPEL tables in a SPEL work file. EV-tables normally have years as rows and regions as columns. The selection of years and regions is identical for all tables defined at the same time. Tables can be transposed by inverting the rows and columns. Each table is described by a code and, optionally, a descriptive text.

In SPEL the code of an EV-table is normally the code of an element of a SPEL table. In other words, all elements of one EV-table correspond to one (always the same) element of a set of SPEL tables. The number of SPEL tables in the set is equal to the number of elements in the EV-table. E.g. the data for wheat production for the years 80 to 85 and the regions D, F and NL form an EV-table, where the source data is obtained from 18 (= 6 years * 3 regions) SPEL tables.

The selection of STE control files may be done using a program selection shell. For further information see 'SPEL System, Technical Documentation (Rev. 1), Vol. 2: BS, SFSS, MFSS'.

6.5.1.2. Types of data files

The EV program supports TAB files and SDA files. TAB files should be preferred because reading and writing is faster.

TAB files:

> SPEL-Data to be processed are kept in TAB files containing various tables. They are distinguished by table types according to data source, scope and treatment.

SDA files:

SDA files may contain both SPEL-coded data not stored in a TAB file and non-SPEL coded data.

6.5.1.3. Control language structure

The available commands are divided into 'main commands' and 'subcommands'. Subcommands can be used only when the corresponding main command was the last main command entered. Main commands have the form '$.....', without any additional instructions.

The subcommands begin with a fixed key word followed by additional parameters separated by at least one blank.

Text parameters containing blanks must be enclosed in quotes.

An EV-session is normally structured as follows:

Definition of general control parameters:

General control parameters e.g. selection of regions, first and last year of time series, type of SPEL tables to be processed etc. are defined (See subcommands of $CONTROL).

Definition of EV-variables (tables) :

The variables to be used are defined by specifying variable codes. The variables are handled as 'EV-Tables' (See subcommands of $NEW and $DEFINE).

Data input :

SPEL-Data are read from the TAB work file (See subcommand of $MATRIX). Other data are read from SDA files or embedded in the EV commands (See $DATA subcommands).

Data treatment :

Data treatment generally affects whole EV-tables. The tables are identified by a single number equal to its position in the sequence of table definition statements. Various arithmetical operations are offered to convert the data into the required form for preparing the printer output or the estimates (See $ARITH subcommands). $REGRESS subcommands are used for regression calculations. The variables to be entered in the regression may be transformed to fit several curve forms.

Data output :

The original data and results can be shown on screen or printer ($TAB fileOUT) or stored in either a TAB file file or in a sequential SDA file.

6.5.1.4. Control language details

Main command for general parameter definition

$CONTROL	Set and list general control parameters

Main commands for on-line help and program flow control

$HELP	On-line help for main commands
$SKIPON	Enable skipping of the following command statements (until the next $SKIPOFF occurs)
$SKIPOFF	Disable skipping of command statements
$STOP	Exiting from EV-session

Main commands for defining variables

$NEW	Definition of new tables (losing the old one)
$DEFINE	Definition of additional tables
$DEFOUT	Current list of table definitions

Main commands for treatment of data

$MATRIX	Process TAB files
$DATA	Process SDA files and non-arithmetical data treatment
$ARITH	Arithmetical operation with tables
$REGRESS	Multiple linear regression analysis with optional use of a forward stepwise algorithm
$TABOUT	List or print tables
$COMMENT	Print comments

$CONTROL Set and list general control parameters

$CONTROL subcommands are used to define general parameters for the EV-session.

syntax: $CONTROL
e.g. $CONTROL

$CONTROL subcommands :

INCLUDE	Define regions to be processed
FROMYEAR	Define first year to be processed
TOYEAR	Define last year to be processed
BASEYEAR	Define base year
TYPIN	Define type codes of tables for data input
TYPOUT	Define type codes of tables for data output
CALL	Call external command files
STATUS	List the status of definitions that may be done by $CONTROL subcommands
ECHOON	Enable echoing of control statements in the log file
ECHOOFF	Disable echoing of control statements in the log file
ERRON	Enable writing protection for the TAB file
ERROFF	Disable writing protection for the TAB file
HELP	On line help for $CONTROL subcommands

INCLUDE: Definition of region selection ($CONTROL subcommand)

Region selection (e.g. Member States) by regional codes.

syntax: INCLUDE *state1 state2 .... state18*
e.g. INCLUDE D F I

The following regional codes can be used at the SPEL/EU level:

D	DEUTSCHLAND	L	LUXEMBOURG
F	FRANCE	UK	UNITED KINGDOM
I	ITALIA	IRL	IRELAND
E	ESPAÑA	DK	DANMARK
P	PORTUGAL	GR	ELLAS
NL	NEDERLAND	E09	EUROP.COMM.(EUR 09)
B	BELGIQUE/BELGIE	E10	EUROP.COMM.(EUR 10)
BL	BEBL/BLEU	E11	EUROP.COMM.(EUR 11)
		E12	EUROP.COMM.(EUR 12)

After changing the regional selection, all data in the program are lost, but the current table definitions are kept.

FROMYEAR: Definition of first year to be processed ($CONTROL subcommand)

First calendar year of calculation period.

syntax: FROMYEAR *year*
e.g. FROMYEAR 1967

TOYEAR: Definition of last year to be processed ($CONTROL subcommand)

Last calendar year of calculation period.

syntax: TOYEAR *year*
e.g. TOYEAR 1986

BASEYEAR: Definition of base year ($CONTROL subcommand)

Base year, e.g. for calculating an index (See $ARITH).

syntax: BASEYEAR *year*
e.g. BASEYEAR 1980

The year must be inside the calculation period FROMYEAR to TOYEAR.

TYPIN: Definition of type code of table for data input ($CONTROL subcommand)

Type code for the SPEL tables from which the data are to be selected. This table type will be used as long as no other type definitions are given (See also $MATRIX and $DATA subcommands).

syntax: TYPIN *type*
e.g. TYPIN CONB

TYPOUT: Definition of type code of table for data output ($CONTROL subcommand)

Type code for the SPEL tables to be stored. This table type will be used, as long as no other type definitions are given (See also $MATRIX and $DATA subcommands).

syntax: TYPOUT *type*
e.g. TYPOUT TESB

CALL: Execution of a STE file ($CONTROL subcommand)

Often repeated command sequences can be stored as macros in additional STE control files. The CALL command can be used to call up the files and execute the commands.

syntax: CALL *controlfilename*
e.g. CALL MACRO1 STE A

STATUS: List the status of definitions ($CONTROL subcommand)

Listing of the current program status (e.g. general control parameters).

syntax: STATUS
e.g. STATUS

ECHOON: Enable echoing of control statements in the log file ($CONTROL subcommand)

All commands entered are logged. Used mainly when working with a control data file to check the command input.

syntax: ECHOON
e.g. ECHOON

ECHOOFF: Disable echoing of control statements in the log file ($CONTROL subcommand)

Discontinues logging of commands.

syntax: ECHOOFF
e.g. ECHOOFF

ERRON: Enable writing protection for the TAB work file ($CONTROL subcommand)

Setting of an internal 'error'-flag; storing of data in the work file is no longer allowed.

syntax: ERRON
e.g. ERRON

Remark : Fatal errors occurring will automatically set the error flag.

ERROFF: Disable writing protection for the TAB work file ($CONTROL subcommand)

See 'ERRON'. ERROFF clears the internal 'error' flag.

syntax: ERROFF
e.g. ERROFF

HELP : On-line help for $CONTROL subcommands ($CONTROL subcommand)

List $CONTROL subcommands, for cursor selection.

syntax: HELP
e.g. HELP

$HELP : On-line help for main commands

This command displays all main commands for cursor selections (See 'List of main commands').

syntax: $HELP
e.g. $HELP

$SKIPON : Enable skipping of the following command statements

By inserting the main command $SKIPON in a control file, the following command statements are skipped. Only the main commands $STOP and $SKIPOFF are interpreted. The command is normally used to de-activate a command sequence in a control file without removing the de-activated commands.

syntax: $SKIPON
e.g. $SKIPON

$SKIPOFF : Disable skipping of the following command statements

See $SKIPON. The command $SKIPOFF disables the skipping function.

syntax: $SKIPOFF
e.g. $SKIPOFF

$STOP: Exit from EV-session

Enter $STOP to finish your EV-session. You will come back to the level from where you started the EV-session.

syntax: $STOP
e.g. $STOP

$NEW: Definition of new tables losing the old ones

To define EV-tables, the variable definition modus must first be called up by entering $NEW or $DEFINE. Then the variables are specified by giving their codes line by line. Later operations then

refer to the variable (table) number rather than to the variable name. The variable number is equal to the position of the table in the sequence of table definitions.

A $NEW command deletes all tables defined so far and starts a new definition sequence.

syntax: $NEW
e.g. $NEW

Until a new $-command occurs, the following lines are subcommands or table definitions. If a line is not a legal subcommand, it is interpreted as a table definition.

$DEFINE: Definition of additional tables keeping the old ones

A $DEFINE command starts a definition sequence to append new tables to the current table definitions.

syntax: $DEFINE
e.g. $DEFINE

Until a new $-command occurs, the following lines are subcommands or table definitions. If a line is not a legal subcommand, it is interpreted as a table definition.

Table definition lines of $NEW or $DEFINE

syntax: xxxxyyyy 0 *descriptiontext*
e.g. WHEAWHEA 0 'Yield coefficient wheat kg/ha'

Up to 8 characters are allowed for defining a table code xxxxyyyy. Names of legal subcommands (See below) are not allowed. Using SPEL codes, the first four characters xxxx specify a column of SPEL tables, the remaining four characters yyyy a row of SPEL tables. Non-SPEL codes are allowed too, but data for those codes cannot be read from SPEL TAB file files.

Currently the '0' must be specified as a 'dummy' for further extensions. A description text could be assigned otherwise the system gets the systems default description text (if available).

Each table definition line excepts an immediately following RENAME subcommand that increments the number of defined tables by one. After creating a new table, the current value of the table counter is implicitly assigned as an identifier to the table defined. This 'table number' represents the position in the sequence of defined tables and is used in all EV operations to specify selected tables. Therefore the table code may not be unique.

Subcommands of $NEW or $DEFINE :

TREND	Definition Trend
RENAME	Rename a defined table
HELP	On-line help for $NEW or $DEFINE subcommands

TREND: Definition of a trend table ($NEW or $DEFINE subcommand)

This subcommand defines a trend table to be used for regression calculations. The table is filled with trend numbers which are automatically generated.

syntax: TREND *startnumber*
e.g. TREND 65

The argument *startnumber* gives the number to be used for the first year in the tables. Trend numbers for the following years are generated using the increment 1. If *startnumber* is missing, 1 is assumed.

RENAME: Rename an already defined table ($NEW or $DEFINE subcommand)

A RENAME command allows the code and description text of a table to be changed. The specified table is replaced by the subsequently defined table.

syntax: RENAME *table*
e.g. RENAME 2

The new code and descriptive text for the specified table must be defined in a 'table definition line' (See above) which must be followed immediately by the RENAME command. This definition line does not increment the table counter.

By entering RENAME without specifying a *table* you will rename the table which was defined last.

HELP: On-line help for $NEW or $DEFINE subcommands $NEW or $DEFINE subcommand)

List of subcommands for cursor selection.

syntax: HELP
e.g. HELP

$DEFOUT: List of current table definitions

List the definitions of tables currently defined. $DEFOUT allows the codes, sequence numbers and descriptive text to be obtained and checked.

syntax: $DEFOUT
e.g. $DEFOUT

Remark : There are no subcommands related to $DEFOUT.

$MATRIX: Reading and writing data from / into TAB file files

This command defines all input and output related to TAB files.

syntax: $MATRIX
e.g. $MATRIX

$MATRIX subcommands

TAB	Definition of work file to be used
SELECT	Definition of a selection of tables to be read and written in
TYPIN	Definition of a type overlay sequence for SPEL tables to be read
GEN	Definition of model area subkey for SPEL tables to be read or written in

MATIN	Reading data
MATOUT	Writing data
MATOUTR	Writing data
CLOSE	Closing access to a work file
HELP	On-line help for subcommands of $MATRIX

TAB: Definition of work file to be used ($MATRIX subcommand)

The TAB subcommand opens a TAB work file either for input only or for both output and input.

syntax: TAB file *iomode filename*
e.g. TAB file INPUT SPEL-USR TAB file A

Legal values for *iomode* are 'INPUT' or 'OUTPUT'. If 'INPUT' is specified, no subsequent writing is possible in the file. If 'OUTPUT' is specified, reading from and writing in the file is allowed.

Only one TAB file can be open at one time. A subsequent TAB file command will automatically close the currently opened file first and then open the currently specified file.

SELECT: Definition of a selection of tables ($MATRIX subcommand)

The subcommand SELECT defines a selection of EV-tables for subsequent reading and writing by MATIN and MATOUT.

syntax: SELECT *table1 ..... table18*
e.g. SELECT 3 6 8

The parameters *table1 .... table18* are EV-table numbers (See 'Definition of Tables').

All table selections are lost when the next main command is entered. The default selection depends on the related subcommand (See MATIN, MATOUT).

GEN: Definition of the model area subkey ($MATRIX subcommand)

The subcommand GEN defines the model area subkey of SPEL tables to be read or written in.

syntax: GEN *area*
e.g. GEN S

TYPIN: Definition of a type overlay sequence for SPEL tables to be read ($MATRIX subcommand)

The subcommand defines an overlay sequence of type subkeys of SPEL tables to read later by the MATIN command.

syntax: TYPIN *type1 type2 .... type18*
e.g. TYPIN ZPAB PRAB

The arguments *type1 type2 .... type18* specify the type subkeys in descending priority from left to right. If no arguments are given, the current overlay definition is deleted. Otherwise the definition is kept for the whole EV session until it is redefined by a new TYPIN subcommand of $MATRIX.

MATIN: Reading of data ($MATRIX subcommand)

MATIN reads data respecting the current selections done by the $CONTROL-subcommands INCLUDE, FROMYEAR, TOYEAR and TYPIN, the EV-table definitions of $NEW or $DEFINE and also respecting the selections by the $MATRIX-subcommands SELECT, TYPIN and GEN. If no SELECT-definitions exist, all defined tables are implicitly selected (See $DEFINE). The reading is done from the file specified in the last TAB file subcommand of $MATRIX.

To read data by MATIN, the codes of all selected tables must also be legal codes of SPEL table elements. EV-table codes which are not SPEL table element codes are not affected by MATIN.

syntax: MATIN *tabletype region1 region2 .... region17*
e.g. MATIN CONB 2 5

The parameter *tabletype* specifies the type subkey of the SPEL tables to be read. If it is missing, the current setting of the TYPIN (See above) is used.

Parameters *region1 ... region17* may define a selection of regions to be read. These parameters specify sequence numbers rather than regional codes. The above example will read SPEL tables of type code CONB for two regions defined as the 2nd and 5th arguments of the INCLUDE subcommand of $CONTROL. If no *region..* parameter is given, data for all defined regions are read.

MATOUT: Writing of data ($MATRIX subcommand)

MATOUT writes data respecting the current selections done by the $CONTROL-subcommands INCLUDE, FROMYEAR, TOYEAR and TYPOUT, the EV-table definitions of $NEW or $DEFINE and respecting also the selections done by the $MATRIX-subcommands SELECT and GEN. If no SELECT-definitions exist, only the 'depending' table of the last regression estimation is implicitly selected. All elements of SPEL tables not currently selected are kept. The file specified in the last TAB file command is written in. The file must be opened for output.

To write data by MATOUT, the code of the selected tables must also be legal codes of SPEL table elements.

syntax: MATOUT *tabletype region1 region2 .... region17*
e.g. MATOUT CONB 2 5

The parameter *tabletype* specifies the type subkey of the SPEL tables to be written. If it is missing, the current setting of TYPOUT (See $CONTROL) is used.

For parameters *region...* see description of subcommand MATIN.

MATOUTR: Writing of data ($MATRIX subcommand)

Same as the subcommand MATOUT, but all elements of SPEL tables not currently selected are reset to zero (0.0).

CLOSE: Close the work file currently open($MATRIX subcommand)

The CLOSE subcommand may be used to update the internal file directory of a work file but does not really close the file access. Further read or write operations may be done without reopening the file.

syntax: CLOSE
e.g. CLOSE

If CLOSE is not explicitly called, the file directory of open work files is updated when opening a new work file by using the TAB subcommand of $MATRIX or by ending the EV session.

HELP: On line help for $MATRIX subcommands (subcommand of $MATRIX)

List of subcommands for cursor selection.

syntax: HELP
e.g. HELP

$DATA: Processing of SDA files and non-arithmetical data treatment

$DATA subcommands allow processing of sequential SDA standard files and general non-arithmetical treatment of data.

syntax: $DATA
e.g. $DATA

$DATA subcommands

STDFILE	Opening of SDA standard files
TYPIN	Definition of a type overlay sequence for SPEL tables to be read
YEARS	Definition of the time period for output
SELECT	Definition of a selection of tables for reading, writing in and 'lagging'
DATIN	Reading of data
DATOUT	Writing of data
LAG	Definition of time lags
TRANSPO	Transposing of current data
DELETE	Deletion of tables
EDIT	Editing of data

STDFILE: Definition of SDA files to be used ($DATA subcommand)

The subcommand STDFILE opens a SDA standard file either for input only or for output only.

syntax: STDFILE *iomode filename*
e.g. STDFILE INPUT MYFILE SDA A

Legal values for *iomode* are 'INPUT' or 'OUTPUT'. The INPUT mode is directly related to the DATIN command, the OUTPUT mode is related to the DATOUT command.

Two different files may be open at one time, one for input and one for output. Any subsequent STDFILE command first will implicitly close the currently opened file with the same *iomode*.

TYPIN: Definition of a type overlay sequence for SPEL tables to be read ($DATA subcommand)

This subcommand defines an overlay sequence of type subkeys of SPEL tables to be read later by the DATIN command.

syntax: TYPIN *type1 type2 .... type18*
e.g. TYPIN EEEB FFFB

The arguments *type1 type2 .... type18* specify the type subkeys in descending priority from left to right. If no arguments are given, the current overlay definition is deleted. Otherwise the definition is kept for the whole EV session until it is redefined by a new TYPIN subcommand of $DATA.

SELECT: Definition of a selection of tables ($DATA subcommand)

Data input and output (DATIN and DATOUT) and definitions of time lags (LAG) can be done selectively for specific tables.

syntax: SELECT *table1 table2 .... table18*
e.g. SELECT 3 5

The parameters *table1 .... table18* are table numbers (See 'Definition of Tables').

All table selections are lost by entering the next main command. The default selection depends on the related subcommand (See MATIN, MATOUT, LAG).

GEN: Definition of the model area subkey ($DATA subcommand)

The subcommand GEN defines the model area subkey of SPEL tables to be read or written in.

syntax: GEN *area*
e.g. GEN S

DATIN: Reading of data ($DATA subcommand)

DATIN reads data respecting the current selections done by the $CONTROL-subcommands INCLUDE, FROMYEAR, TOYEAR and TYPIN, the EV-table definitions of $NEW or $DEFINE and respecting also the selections by the $DATA-subcommands SELECT, TYPIN and GEN. If no SELECT-definitions exist, all defined tables (See $DEFINE) are implicitly selected. The reading is done from the file opened in the last STDFILE command with INPUT mode specified.

To read data by DATIN, the table codes used in the file to be read must match the table codes defined in the EV session.

syntax: DATIN *tabletype region1 region2 ... region17*
e.g. DATIN XXXB 2 5

The parameter *tabletype* specifies the type subkey of the SPEL tables to be read. If it is missing, the current setting of the TYPIN (See above) is used.

Parameters *region1 ... region17* may define a selection of regions to be read. These parameters specify sequence numbers rather than regional codes. The above example will read SPEL tables of type code CONB for two regions defined as the 2nd and 5th argument of the INCLUDE subcommand of $CONTROL. If no *region..* parameter is given, data for all defined regions are read.

DATOUT: Writing of data (subcommand of $DATA)

DATOUT writes data respecting the current selections done by the $CONTROL-subcommands INCLUDE, FROMYEAR, TOYEAR and TYPOUT, the EV-table definitions of $NEW or $DEFINE and respecting also the selections done by the $DATA-subcommands SELECT and GEN. If no SELECT-definitions exist, only the 'depending' table of the last regression estimation is implicitly selected. Writing is done into the file which was opened for OUTPUT mode.

To write data by MATOUT, the codes (See $DEFINE or $NEW above) of the selected EV-tables must also be legal SPEL table element codes.

syntax: MATOUT *tabletype region1 region2 .... region17*
e.g. MATOUT YYYB 2 5

The parameter *tabletype* specifies the type subkey of the SPEL tables to be written. If it is missing, the current setting of TYPOUT (See $CONTROL) is used.

For parameters *region...* see description of DATIN subcommand.

YEARS: Definition of time period for output ($DATA subcommand)

Using the subcommand YEARS the time series written by the next DATOUT command may be shorter than the calculation period defined by $CONTROL-subcommands FROMYEAR and TOYEAR.

syntax: YEARS *firstyear lastyear*
e.g. YEARS 70 75

The parameters *firstyear* and *lastyear* represent the first and last year of data which will be output by subsequent DATOUT commands. The output period is kept for the whole EV session until it is redefined by a new YEARS subcommand of $DATA. If the new defined output period exceeds the calculation period it is clipped.

LAG: Definition of time lags ($DATA subcommand)

The LAG subcommand offers 'permanent lagging' of time series in the EV core memory. Subsequently all input, output and arithmetical commands will process the 'lagged' data. For temporary 'lagging' of dependent and independent regression analysis variables see the main command $REGRESS.

The LAG subcommand must always be used in combination with the SELECT subcommand.

syntax: SELECT *table1 table2 .... table18*
LAG *timelag1 timelag2 .... timelag18*

e.g. SELECT 2 3
LAG 1 2

The LAG arguments correspond to the SELECT arguments. In the example, *timelag2* defines the lag shift for the table specified by *table2*.

All elements t of time series X are shifted as: $X_t = X_{(t+timelag)}$, where *timelag* must be a positive or negative integer. Depending on this number, at one end of the time series data figures are lost and at the other end zero figures are inserted.

TRANSPO: Transposing the current data ($DATA subcommand)

In the normal status the data in the EV-tables are organized as

Table rows = years
Table columns = regions
Table headers = variables

Transposing data organized normally will change the organization to

Table rows = years
Table columns = variables
Table headers = regions

Transposing again will re-establish the normal status.

syntax: TRANSPO
e.g. TRANSPO

DELETE: Deletion of tables ($DATA subcommand)

The subcommand DELETE deletes EV-tables.

syntax: DELETE *table1 table2 .... table18*
e.g. DELETE 2 3 4

The parameters *table1 .... table18* are table numbers (See 'Definition of Tables').

Remark : When deleting tables which are not at the end of the definition sequence, the remaining table numbers will change. Assuming there were six originally defined tables, the deletion example will rename the remaining original table numbers 1, 5 and 6 as new table numbers 1, 2 and 3.

EDIT: Editing of data ($DATA subcommand)

The subcommand EDIT allows data in the EV-tables to be changed or new data to be inserted into the tables. Editing is always related to a column of an EV-table. To edit data in a table column, the subcommand EDIT first specifies the table number and the column number.

syntax: EDIT table column
e.g. EDIT 2 3

Immediately following the EDIT command, the data are entered in one or more 'data lines'.

syntax: *year value*
e.g. 70 135.5

Editing of a table column must be terminated by entering a blank line. To edit another table or column, the complete edit sequence must be repeated.

$ARITH: Arithmetical calculations

Various arithmetical and transformation operations for processing data are offered by the subcommands of $ARITH. All operations affect complete EV-tables which are specified as command operands.

syntax: $ARITH
e.g. $ARITH

Subcommands of $ARITH

ADD	Addition of table elements to table elements
SUB	Subtraction of table elements from table elements
MUL	Multiplication of table elements by table elements
DIV	Division of table elements by table elements
ADDC	Addition of constants to table elements
SUBC	Subtraction of constants from table elements
MULC	Multiplication of table elements by constants
DIVC	Division of table elements by constants
ROOTC	Root of table elements
POTC	Raising to a power of table elements
LN	Natural logarithm of table elements
LOG	Logarithm (base 10) of table elements
REZ	Reciprocal of table elements
EXP	Exponential operation for table elements
POW	Power operation for table elements
INDEXGR	Conversion of growth rate table elements to indices
GROWTH	Growth rates in percent for table elements
BASIS	Division of table elements by base year elements of the same tables
MULB	Multiplication of table elements by base year elements of other tables
DIVB	Division of table elements by base year elements of other tables
ADDB	Addition of base year elements of other tables to table elements
SUBB	Subtraction of base year elements of other tables from table elements
MAVC	Replacement of table elements by 'Moving averages'
RESET	Resetting of table elements
CLEAN	Resetting of table elements depending on reference tables
OVERLAY	Replacement of zero table elements with elements of other tables
HELP	On-line help for subcommands of $ARITH

Results of arithmetical operations change only the first operand or all operands depending on the command.

ADD: Addition of table elements to table elements ($ARITH subcommand)

syntax: ADD *table1 table2 .... table9*
e.g. ADD 5 2

Operation: $table1_{t,c} = table1_{t,c} + table2_{t,c} + ... + table9_{t,c}$

SUB: Subtraction of table elements from table elements ($ARITH subcommand)

syntax: SUB *table1 table2 .... table9*
e.g. SUB 5 2

Operation: $table1_{t,c} = table1_{t,c} - table2_{t,c} - .... - table9_{t,c}$

MUL: Multiplication of table elements by table elements ($ARITH subcommand)

syntax: MUL *table1 table2 .... table9*
e.g. MUL 5 2

Operation: $table1_{t,c} = table1_{t,c} * table2_{t,c} * .... * table9_{t,c}$

DIV: Division of table elements by table elements ($ARITH subcommand)

syntax: DIV *table1 table2 .... table9*
e.g. DIV 5 2

Operation: $table1_{t,c} = \frac{table1_{t,c}}{table2_{t,c} * table3_{t,c} * .... * table9_{t,c}}$

ADDC: Addition of constants to table elements ($ARITH subcommand)

syntax: ADDC *table1 constant1 .... constant8*
e.g. ADDC 5 2.55

Operation: $table1_{t,c} = table1_{t,c} + constant1 + constant2 + ... + constant8$

SUBC: Subtraction of constants from table elements ($ARITH subcommand)

syntax: SUBC *table1 constant1 .... constant8*
e.g. SUBC 5 2.55

Operation: $table1_{t,c} = table1_{t,c} - constant1 - constant2 - .... - constant8$

MULC: Multiplication of table elements by constants ($ARITH subcommand)

syntax: MULC *table1 constant1 .... constant8*
e.g. MULC 5 2.55

Operation: $table1_{t,c} = table1_{t,c} * constant1 * constant2 * .... * constant8$

DIVC: Division of table elements by constants ($ARITH subcommand)

syntax: DIVC *table1 constant1 .... constant8*
e.g. DIVC 5 2.55

Operation: $table1_{t,c} = \frac{table1_{t,c}}{constant1 * constant2 * .... * constant8}$

ROOTC: Root of table elements ($ARITH subcommand)

syntax: ROOT *table1 constant1*
e.g. ROOT 5 4.

Operation: $table1_{t,c} = \sqrt[constant1]{table1_{t,c}}$

POTC: Raising to a power of table elements ($ARITH subcommand)

syntax: POTC *table1 constant1*
e.g. POTC 5 4.

Operation: $table1_{t,c} = table1_{t,c}^{constant1}$

LN: Natural logarithm of table elements ($ARITH subcommand)

syntax: LN *table1 table2 .... table9*
e.g. LN 1 3 6

Operation: $table_{i,t,c} = LOG_e(table_{i,t,c})$

LOG: Logarithm (base 10) of table elements ($ARITH subcommand)

syntax: LOG *table1 table2 .... table9*
e.g. LOG 1 3 6

Operation: $table_{i,t,c} = LOG_{10}(table_{i,t,c})$

REZ: Reciprocal of table elements ($ARITH subcommand)

syntax: REZ *table1 table2 .... table9*
e.g. REZ 1 3 6

Operation: $table_{i,t,c} = \frac{1}{table_{i,t,c}}$

EXP: Exponential operation for table elements ($ARITH subcommand)

syntax: EXP *table1 table2 .... table9*
e.g. EXP 1 3

Operation: $table_{i,t,c} = e^{table_{i,t,c}}$

POW: Power operation for table elements (subcommand of $ARITH)

syntax: POW *table1 table2 .... table9*
e.g. POW 1 3 6

Operation: $table_{i,t,c} = 10.^{table_{i,t,c}}$

INDEXGR: Conversion of growth rate table elements to indices ($ARITH subcommand)

syntax: INDEXGR *table1 table2 .... table9*
e.g. INDEXGR 1 3 6

Operation: $table_{i,1,c} = 100.$

$$table_{i,t,c} = \left(1. + \frac{table_{i,t-1,c}}{100.}\right) \qquad \text{for } t > 1$$

GROWTH: Growth rates in percent for table elements ($ARITH subcommand)

syntax: GROWTH *table1 table2 .... table9*
e.g. GROWTH 1 3 6

Operation: $table_{i,1,c} = 0.$

$$table_{i,t,c} = \left(\frac{table_{i,t,c}}{table_{i,t-1,c}} - 1.\right) * 100. \qquad \text{for } t > 1$$

BASIS: Division of table elements by base year elements of the same table ($ARITH subcommand)

syntax: BASIS *table1 table2 .... table9*
e.g. BASIS 1 3 6

Operation: $$table_{i,t,c} = \frac{table_{i,t,c}}{table_{i,tb,c}}$$

where *tb* points to the base year defined by the BASEYEAR subcommand of $CONTROL

Remark : To get an index-time series based on 100, use the subcommand MULC to multiply the results by 100.
e.g. MULC 1 100.
MULC 3 100.
MULC 6 100.

MULB: Multiplication of table elements by base year elements of other tables ($ARITH subcommand)

syntax: MULB *table1 table2 .... table9*
e.g. MULB 1 3 6

Operation: $table1_{t,c} = table1_{t,c} * table2_{tb,c} * table3_{tb,c} * .... * table9_{tb,c}$

where *tb* points to the base year defined by the BASEYEAR subcommand of $CONTROL

DIVB: Division of table elements by base year elements of other tables ($ARITH subcommand)

syntax: DIVB *table1 table2 .... table9*
e.g. DIVB 1 3 6

Operation: $table1_{t,c} = \frac{table1_{t,c}}{table2_{tb,c} * table3_{tb,c} * .... * table9_{tb,c}}$

where *tb* points to the base year defined by the BASEYEAR subcommand of $CONTROL

MAVC: Replacement of table elements with 'Moving averages' ($ARITH subcommand)

syntax: MAVC *table n*
e.g. MAVC 3 3

Operation: $table_{t,c} = \frac{\sum_{p=1}^{n} table_{t+p-h-1,c}}{n}$

where *h* is the integer part of $\frac{n}{2}$

Remark : The formula implies time subscripts of less than 1 at the beginning of time series as well as time subscripts greater than the number of years defined at the end of time series. The related data elements are generated by duplicating the next available element.

RESET: Resetting of table elements ($ARITH subcommand)

syntax: RESET *table1 table2 .... table9*
e.g. RESET 1 3 6

Operation: $table_{i,t,c} = 0.$

CLEAN: Resetting of table elements depending on reference tables ($ARITH subcommand)

syntax: CLEAN *table1 table2 .... table9*
e.g. CLEAN 1 3 6

Operation: Changes $table1_{t,c}$ to zero if the corresponding element in at least one of *table2 .... table9 is also zero.*

OVERLAY : Replacement of zero table elements with elements of other tables ($ARITH subcommand)

syntax: OVERLAY *table1 table2 .... table9*
e.g. OVERLAY 1 3 6

Operation: Replace zero elements of $table1_{t,c}$ with corresponding elements of *table2 .... table9* until $table1_{t,c}$ is no longer zero.

HELP: On-line help for $ARITH subcommands ($ARITH subcommand)

List of subcommands for cursor selection.

syntax: HELP
e.g. HELP

$REGRESS : Multiple linear regression analysis with optional use of a forward stepwise algorithm

$REGRESS subcommands offer linear regression analysis with one or more independent variables. The subcommands affect complete EV-tables which are specified as command operands. All currently defined EV-tables may be defined as the regression variables. The variables to be entered into the regression are specified as table numbers (See $DEFINE).

The variables can be transformed to fit several regression curves. Separate estimates are made for each table column. Optional step-by-step entering of independent variables may be selected.

EV calculates the usual statistical fit parameters (e.g. Durbin Watson coefficient, coefficient of determination, etc.). The 'correlation coefficients' of regression equations and common 'elasticities' are also available for each estimate.

syntax: $REGRESS
e.g. $REGRESS

Subcommands:

SELECT	Definition of variables to be entered in the regression
ESTIMATE	Regression estimation with optional variable transformations
PCTNEW	Definition of variance threshold for step-by-step entering of variables
LISTON	Enable listing of correlation coefficients and elasticities
LISTOFF	Disable listing of correlation coefficients and elasticities
PRINTON	Enable printing of correlation coefficients and elasticities
PRINTOFF	Disable printing of correlation coefficients and elasticities

SELECT: Definition of variables to be entered in the regression ($REGRESS subcommand)

syntax: SELECT *table1 table2 .... table18*
e.g. SELECT 1 3 6

The first operand *table1* is defined as the dependent variable and the remaining operands *table2 .... table18* are defined as independent variables. The variable selection is valid only for the next ESTIMATE subcommand. ESTIMATE will re-establish the default selection which is equal to the general variable definition (See $NEW and $DEFINE).

ESTIMATE: Regression estimation with optional variable transformations ($REGRESS subcommand)

syntax: ESTIMATE *trans1 trans2 .... trans18*
e.g. ESTIMATE LIN LOG LOG

The operands *trans1 trans2 .... trans18* define transformation codes for the corresponding selected regression variables (e.g. *trans1* is the transformation to be done for *table1)*. New estimated values for the dependent variable are put into an EV-table which is identified by table number 999. Correlation coefficients and elasticities are listed and/or printed only if listing and/or printing is enabled. The usual statistical fit parameters are prepared to permit access by the $TABOUT-subcommands LISTR and PRINTR.

The following transformations are possible for both dependent and independent variables:

LIN	Linear (no transformation)	$x' = x$
LN	Natural logarithm	$x' = LOG_e(x)$
LOG	Logarithm base 10	$x' = LOG_{10}(x)$
SQR	Square root	$x' = \sqrt{x}$
REZ	Reciprocal	$x' = \frac{1}{x}$
EXP	Exponential	$x' = e^x$
POW	Power	$x' = 10^x$
QUA	Square	$x' = x^2$

where x is an element of tables to be transformed.

PCTNEW: Definition of variance threshold for stepwise entering of variables ($REGRESS subcommand)

syntax: PCTNEW *pct*
e.g. PCTNEW 100.

'Stepwise' regression is allowed only when *pct* is not equal to zero. On default, the 'stepwise' option is disabled (*pct* = zero). The definition is kept for the whole EV-session until it is redefined by a new PCTNEW command.

The parameter *pct* defines a threshold for the ratio (in °/₀₀) of explanation rate of the variance of each independent variable related to the total variance. A dependent variable is entered in the regression when the explanation rate is greater or equal *pct.*

LISTON: Enable listing of correlation coefficients and elasticities ($REGRESS subcommand)

syntax: LISTON
e.g. LISTON

Listings of coefficients are displayed by the following ESTIMATE commands.

LISTOFF: Disable listing of correlation coefficients and elasticities ($REGRESS subcommand)

syntax: LISTOFF
e.g. LISTOFF

(See LISTON)

PRINTON: Enable printing of correlation coefficients and elasticities ($REGRESS subcommand)

syntax: PRINTON
e.g. PRINTON

Listings of coefficients are printed by the following ESTIMATE commands.

PRINTOFF: Disable printing of correlation coefficients and elasticities ($REGRESS subcommand)

syntax: PRINTOFF
e.g. PRINTOFF

(See PRINTON)

$TABOUT: Listing or printing of tables

The figures in normal EV-tables defined by $NEW or $DEFINE and the figures in special tables constructed by $REGRESS are displayed or printed.

syntax: $TABOUT
e.g. $TABOUT

Like other SPEL programs, EV prints in a 'printer file' (e.g. named EV LST) rather than sending directly to the printer. The 'printer file' may be printed after the end of the EV-session.

Subcommands:

LISTD	List normal tables on the screen
PRINTD	Print normal tables
LISTR	List regression result tables on the screen
PRINTR	Print regression result tables
COMMENT	Print comments
PRINTON	Enable subsequent printing of tables
PRINTOFF	Disable subsequent printing of tables
PAGE	Define printer page height
RECORD	Define printer page width
CNUMON	Enable column numbering
CNUMOFF	Disable column numbering
ROWS	Select table rows to be printed
LEGON	Enable printing of table legends
LEGOFF	Disable printing of table legends

LISTD: List normal tables on the screen ($TABOUT subcommand)

syntax: LISTD *table1 table2 .... table18*
e.g. LISTD 1 3 6

The arguments *table1 table2 .... table18* specify the numbers of tables to be listed. If no parameters are specified, all tables are listed.

The meaning of headers and columns depends on the current transposition mode (See the TRANSPO-subcommand of $DATA). Rows are always reserved for years.

The following additional rows of simple statistical figures are generated:

MEAN	Arithmetical mean of all years
GROWTH%	Arithmetical mean of annual growth rates in percent
G.GR.%	Geometrical mean of annual growth rates in percent
C.OF VAR	Coefficient of variation in percent

Zero figures are excluded from the computation of the statistical figures.

PRINTD: Print normal tables ($TABOUT subcommand)

syntax: PRINTD *table1 table2 .... table18*
e.g. PRINTD 1 3 6

(See LISTD)

LISTR: List regression result tables on the screen ($TABOUT subcommand)

syntax: LISTR *rtable1 rtable2 .... rtable18*
e.g. LISTR 1 2 5

The LISTR command is only valid if regression results have been prepared (See the ESTIMATE subcommand of $REGRESS).The arguments *table1 table2 .... table18* specify numbers which identify regression result tables to be listed. If no parameters are specified, all regression result tables are listed.

The following regression tables may be selected:

Table 1:	Estimated figures for the dependent variable
Table 2:	Partial Regression Coefficients
Table 3:	Standard error of regression coefficients
Table 4:	T-values
Table 5:	Error probabilities
Table 6:	Other statistical figures

The meaning of table headers and columns depends on the current transposition mode (See the TRANSPO subcommand of $DATA).

The layout of regression table 1 is as for normal tables (See LISTD). In regression tables 2 to 5 the rows are the variables which are entered into the regression.

The rows of regression table 6 have a special meaning:

CONSTANT	Intercept constant of the regression
DEGRFREE	Degrees of freedom
OBSERVAT	Number of observations
%.SUM.SQ	Percentage of sum of squares of deviations
M.CORREL	Multiple correlation coefficient
M.CORR.A	Multiple correlation coefficient adjusted
STD.ERR.	Standard estimation error
ST.ERR.A	Standard estimation error, adjusted
F-RATIO	F-ratio

DURBIN	Durbin-Watson statistic
THEIL	Theil's coefficient

PRINTR: Print regression result tables ($TABOUT subcommand)

syntax: PRINTR *rtable1 rtable2 .... rtable6*
e.g. PRINTR 1 2 5

(See LISTR)

COMMENT: Print comments ($TABOUT subcommand)

syntax: COMMENT
text
e.g. COMMENT
This is the text printed.

The subcommand allows one line of comment to be printed. The comment *text* is specified by a separate line which must immediately follow the COMMENT command line. Carriage control is supported by special characters entered at the beginning of the comment line (See $COMMENT).

PRINTON: Enable subsequent printing of tables ($TABOUT subcommand)

syntax: PRINTON
e.g. PRINTON

The subcommands may be used to enable printing after it has been disabled by a previous PRINTOFF command.

PRINTOFF: Disable subsequent printing of tables ($TABOUT subcommand)

syntax: PRINTOFF
e.g. PRINTOFF

Subsequent subcommands PRINTD or PRINTR are ignored until printing is enabled again by PRINTON.

PAGE: Definition of printer page height ($TABOUT subcommand)

syntax: PAGE *lines*
e.g. PAGE 67

The default page height is 57 lines.

RECORD: Definition of printer page width ($TABOUT subcommand)

syntax: RECORD *characters*
e.g. RECORD 80

The default page width is 132 characters. Legal values are 72 to 132.

CNUMON: Enable column numbering ($TABOUT subcommand)

syntax: CNUMON
e.g. CNUMON

The command enables printing of column sequence numbers in tables printed by subsequent PRINTD and PRINTR commands. On default only column codes are printed as column headers.

CNUMOFF: Disable column numbering ($TABOUT subcommand)

syntax: CNUMOFF
e.g. CNUMOFF

(See CNUMON)

ROWS: Select table rows to be printed ($TABOUT subcommand)

syntax: ROWS *row1 row2 .... row18*
e.g. ROWS 1 3 5

This subcommand defines a selection of table rows to be respected in subsequent subcommands LISTD and PRINTD. The arguments are relative row numbers related to range defined by the $CONTROL-subcommands FROMYEAR and TOYEAR. If FROMYEAR is set to 1981, the above example will select the rows of years 1981, 1983 and 1985. A table row selection is kept for the whole EV session until it is redefined by a new ROWS command. On default all table rows are selected.

LEGON: Enable printing of table legends ($TABOUT subcommand)

syntax: LEGON
e.g. LEGON

If legend printing is enabled, a column description is printed in addition to the column codes in the header of the tables. On default the legend printing is disabled.

LEGOFF: Disable printing of table legends ($TABOUT subcommand)

syntax: LEGOFF
e.g. LEGOFF

(See LEGON)

$COMMENT: Printing of comments

This command allows comments to be placed in the printer output. After this command is entered the following lines in the command stream are interpreted as comments to be printed until a new $....-command occurs.

If the first character of a comment line is not a blank character, it has been given a special treatment: '0' will insert one blank line before printing the comment. '1' will start a new output page. Each character other than a blank character will be repeated to fill a whole output line. If this occurs, the comment itself will be printed in the next output line.

6.5.1.5. A typical EV session

The structure of the control language reflects the work steps within EV applications. Reading control files of existing applications will help the user to learn the control language.

Example of a typical STE file

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
$CONTROL
INCLUDE F I
FROMYEAR 1973
TOYEAR 1988
TYPOUT ESTB
BASEYEAR 1980
$TABOUT
RECORD 80
LEGON
$COMMENT

        --- TABLE WINE ---
 ESTIMATION BY REGRESSION, PINDTWIN = F(PINDWINT)

$NEW
PINDTWIN 1
PINDWINT 2
$MATRIX
TAB OUTPUT SPEL-USR TAB A
GEN D
TYPIN ZPAB EXPB
MATIN
SELECT 2
TYPIN ZPAC EXPC ESTC
MATIN
$DATA
TRANSPO
$TABOUT
PRINTD
$DATA
TRANSPO
$REGRESS
SELECT 1 2
ESTIMATE LIN
$TABOUT
PRINTR
$MATRIX
SELECT
MATOUT ESTB 1
$STOP
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----
```

The control file results in the following printer output:

```
        --- TABLE WINE ---
 ESTIMATION BY REGRESSION, PINDTWIN = F(PINDWINT)

---------------------------------
TABLE    1: PINDTWIN PRICE INDEX (1980=100);
---------------------------------
                   F          I
---------------------------------
 1 ! 1973 !        0.00       0.00
 2 ! 1974 !        0.00       0.00
 3 ! 1975 !        0.00       0.00
 4 ! 1976 !        0.00       0.00
 5 ! 1977 !        0.00       0.00
 6 ! 1978 !        0.00       0.00
 7 ! 1979 !        0.00       0.00
 8 ! 1980 !        0.00       0.00
 9 ! 1981 !        0.00       0.00
10 ! 1982 !     9280.00   10468.00
11 ! 1983 !     8614.00   22399.00
12 ! 1984 !    11213.00   15881.00
13 ! 1985 !     8891.00   10916.00
14 ! 1986 !     7649.00   22710.00
15 ! 1987 !    12062.00   20724.00
16 ! 1988 !     5800.00   14222.00
---------------------------------
 1 MEAN         9072.71   16760.00
 2 GROWTH %       -0.98      20.26
 3 G.GR.%         -7.53       5.24
 4 C.OF VAR       21.46      28.87
---------------------------------
DATE:   26.04.91    SOURCE: SPEL LUX., 'BDS'-TYPE:  ZPAB
```

```
---------------------------------
TABLE    2: PINDWINT PRICE INDEX (1980=100);
---------------------------------
                    F          I
---------------------------------
 1 ! 1973 !    18642.00    3450.00
 2 ! 1974 !    24887.00    9527.00
 3 ! 1975 !    13575.00    3510.00
 4 ! 1976 !    14970.00    1801.00
 5 ! 1977 !     8557.00    1700.00
 6 ! 1978 !     9276.00    2060.00
 7 ! 1979 !    21485.00    9400.00
 8 ! 1980 !    19748.00   14653.00
 9 ! 1981 !    11871.00   10564.00
10 ! 1982 !    20069.00   10831.00
11 ! 1983 !    16856.00   23153.00
12 ! 1984 !    17800.00   17750.00
13 ! 1985 !    17626.00   12828.00
14 ! 1986 !    16777.00   24815.00
15 ! 1987 !    20652.00   23332.00
16 ! 1988 !    13736.00   15588.00
---------------------------------
 1 MEAN        16657.94   11560.12
 2 GROWTH %        6.00      38.91
 3 G.GR.%         -2.02      10.58
 4 C.OF VAR       25.92      66.23
---------------------------------
DATE:   26.04.91    SOURCE: SPEL LUX., 'BDS'-TYPE:  ZPAC
```

```
---------------------------------
TABLE    3: PINDTWIN  ESTIMATED VALUES
            DEP:LIN   INDEP:LIN
---------------------------------
                    F          I
---------------------------------
 1 ! 1973 !     9836.71    2871.19
 2 ! 1974 !    14622.91    8544.09
 3 ! 1975 !     5953.34    2927.20
 4 ! 1976 !     7022.47    1331.84
 5 ! 1977 !     2107.52    1237.55
 6 ! 1978 !     2658.57    1573.61
 7 ! 1979 !    12015.60    8425.54
 8 ! 1980 !    10684.36   13329.24
 9 ! 1981 !     4647.39    9512.14
10 ! 1982 !    10930.37    9761.38
11 ! 1983 !     8467.91   21264.03
12 ! 1984 !     9191.40   16220.30
13 ! 1985 !     9058.04   11625.59
14 ! 1986 !     8407.37   22815.52
15 ! 1987 !    11377.18   21431.13
16 ! 1988 !     6076.73   14202.07
---------------------------------
 1 MEAN         8316.11   10442.02
 2 GROWTH %       23.28      45.82
 3 G.GR.%         -3.16      11.25
 4 C.OF VAR       39.79      68.45
---------------------------------
DATE:   26.04.91    SOURCE: SPEL LUX., ESTIMATE
```

```
---------------------------------
TABLE    4: PINDTWIN  PARTIAL REGRESSION COEFFICIENTS
            ABH:LIN   UNABH:LIN
---------------------------------
                  F          I
---------------------------------
 1 PINDWINT       0.77       0.93
---------------------------------
DATE:   26.04.91    SOURCE: SPEL LUX., ESTIMATE



---------------------------------
TABLE    5: PINDTWIN  STANDARD ERROR OF REGR.-COEFFICIENTS
            ABH:LIN   UNABH:LIN
---------------------------------
                  F          I
---------------------------------
 1 PINDWINT       0.22       0.06
---------------------------------
DATE:   26.04.91    SOURCE: SPEL LUX., ESTIMATE



---------------------------------
TABLE    6: PINDTWIN  T-VALUES
            ABH:LIN   UNABH:LIN
---------------------------------
                  F          I
---------------------------------
 1 PINDWINT       3.41      16.61
---------------------------------
DATE:   26.04.91    SOURCE: SPEL LUX., ESTIMATE



---------------------------------
TABLE    7: PINDTWIN  ERROR PROBABILITY, %
            ABH:LIN   UNABH:LIN
---------------------------------
                  F          I
---------------------------------
 1 PINDWINT       1.43       0.00
---------------------------------
DATE:   26.04.91    SOURCE: SPEL LUX., ESTIMATE
```

TABLE 8: PINDTWIN INTERCEPT CONSTANT AND TEST STATISTICS
DEP:LIN INDEP:LIN

		F	I
1	CONSTANT	-4450.60	-349.41
2	DEGRFREE	6.00	6.00
3	OBSERVAT	7.00	7.00
4	%.SUM.SQ	69.93	98.22
5	M.CORREL	0.84	0.99
6	M.CORR.A	0.84	0.99
7	STD.ERR.	1263.37	763.91
8	ST.ERR.A	1263.37	763.91
9	F-RATIO	11.63	275.81
10	DURBIN	1.87	1.29
11	THEIL	0.27	0.08

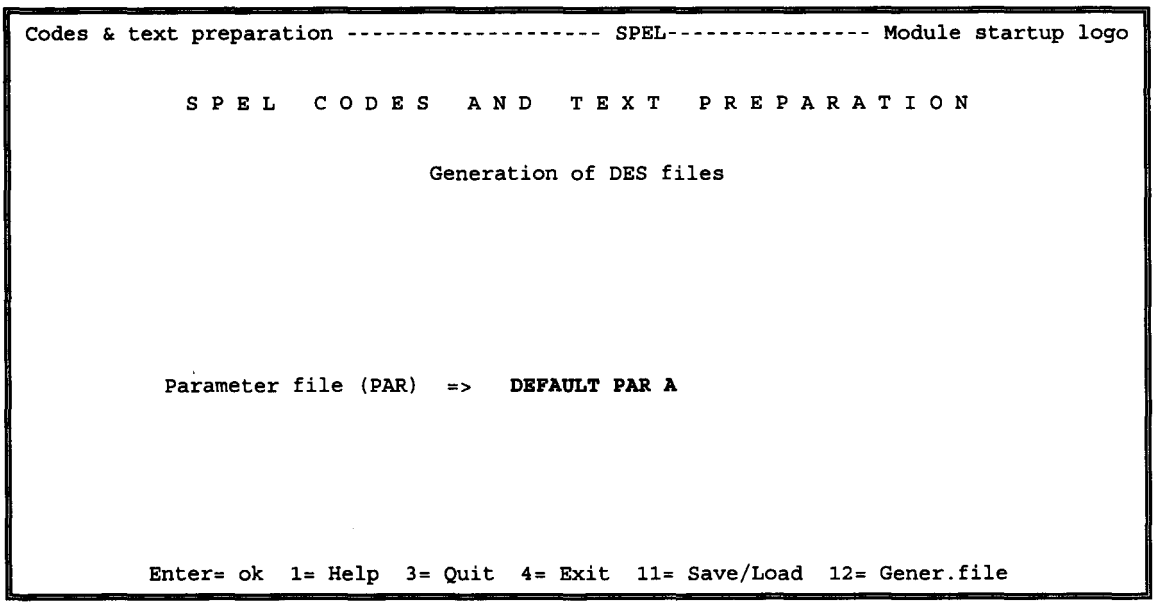
DATE: 26.04.91 SOURCE: SPEL LUX., ESTIMATE

6.6. Description management

6.6.1. DESCRI program: generate code description text

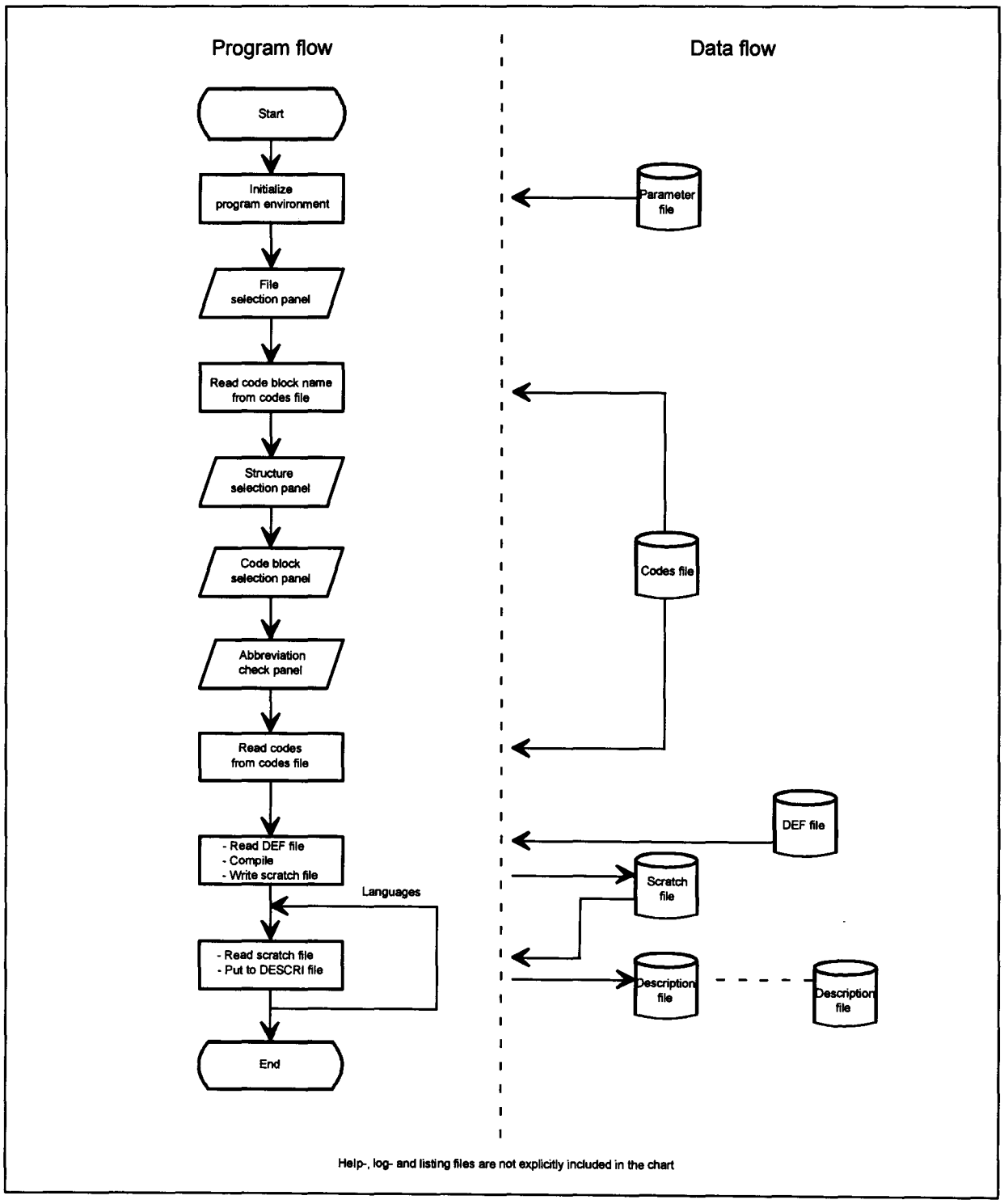
The DESCRI program compiles DEF files which contain description text for codes. The table structures for which the program is to build up description text for table element codes are selected by panels. For all codes fitting the selected table structures, the DESCRI program compiles text description as specified in the DEF file. Each of the resulting descriptions is saved as an entry in a DES file which is a special application of the TAB format. All SPEL programs may use these descriptions as further explanations for codes, e.g. in printed tables.

Module startup screen



For detailed information, see chapter 'General programs'.

Figure 13: DESCRI program flow chart



Work file selection screen

```
Codes & text preparation ----------------- SPEL ---------------------- File selection

                              Please enter file names




 Text input file (DEF)          => SPEL-EU DEF F
 Codes input file (COD)         => SPEL-EU COD F
 Text output file (DES) la.:1 => ENGLISH DES F
 Text output file (DES) la.:2 => GERMAN DES F
 Text output file (DES) la.:3 => FRENCH DES F
 Text output file (DES) la.:4 => ITALIAN DES F




                Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Text input file (DEF)

This DEF input file defines the rules for building description text (see below).

Codes input file (COD)

This COD input file contains the list of column and row codes for one or more SPEL model(s) and one or more table structure(s) (see chapter 'Formal structuring and coding: COD files').

Text output file (DES) la.: ...

The DESCRI program will store the text descriptions as entries in this DES file. For every language, the user has to define a separate output file.

Structure selection screen

```
Codes & text preparation ---------------- SPEL ----------------- Structure selection

                Please enter table structure identifier to describe




 Table structure identifier ( 2 ch.) => SB DB SC DC




                Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

In this panel the user can select the table structure codes for which the DESCRI program is to compile the description text.

Parameters:

Table structure identifier

Enter one table structure identifier or a list of table structure identifiers as follows:

ms [*ms* ....]

m model area, such as 'S' (Supply)

s table structure type. Use 'C' for complementary table structure and 'B' for basic table structure

Code block selection screen

```
Codes & text preparation ---------------- SPEL ---------------- Code block selection

      Please enter code block names of columns and rows for each table structure



 Column code block for structure: SB => CSB
 Row code block for structure   : SB => LSB
 Column code block for structure: DB => CDB
 Row code block for structure   : DB => LDB
 Column code block for structure: SC => CSC
 Row code block for structure   : SC => LSC
 Column code block for structure: DC => CDC
 Row code block for structure   : DC => LDC









                Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Within the COD file, the codes for columns and rows are collected in blocks identified by unique block codes of three characters. In this panel the user links block code names to the table structure type selected in the previous panel.

Parameters:

Column code block for structure : ...

> Enter the block code as used in the COD file to identify the column dimension of the related table structure. In SPEL the block code is build from the table structure identifier with the prefix character 'C'.

Row code block for structure : ...

> Enter the block code as used in the COD file to identify the row dimension of the related table structure. In SPEL the block code is build from the table structure identifier with the prefix character 'L'.

6.6.1.1. DEF files

DEF files are used to define description text for codes. The code description text is useful for all logs, listings and exploitations. The SPEL table element code 'SWHELEVL' will be explained here by the description text 'main area soft wheat 1000 ha'.

The DEF files are sequential files with a fixed record length of 255. For user friendly editing, the line length should not exceed 72 characters.

File structure:

The DEF file is composed of comment lines and statements. There are two types of statements: the DEF statement and the ASS statement. A statement may be continued on continuation lines up to a total statement length of 255 characters. Except inside literals, lower-case characters are treated as if they were the upper case equivalent. The last character of a statement is always a semicolon.

- Comment line

Comment lines are denoted by placing a '*' in the first character column of the line. Comments may occur anywhere in the file; they may be intermingled with statement continuation lines.

- DEF statement

These statements are used to define symbolic text constants for subsequent inclusion in ASS statements.

Syntax of constant definitions:.

DEF S=*symbol*, T=*text*;

- or -

DEF S=*symbol*, T1=*text* **[**,T2=*text*,....**]**;

where:

Parameter S:

symbol Symbolic name for the text constant to be defined. It is always a single blank-delimited word of up to eight characters.

Parameter T.....:

text Text literal (text enclosed in apostrophes) to be assigned to *symbol*. Double apostrophes may be used to enclose single apostrophes inside the text, and vice versa. The enclosing apostrophes inclosing are not part of the datum.

Either the same text (T=*text*) or different text (T1=*text*, T2=....) may be assigned for the different languages 1, 2,...).

For indirect referencing of the symbolic text constants (see description of ASS statements), it may be helpful to add a suffix ('.c' or '.r') to the generic name. When a symbolic constant is indirectly referenced by the reserved name COL, a name with the suffix '.c' has a higher priority than a name without the suffix '.c'. The suffix '.r' is used for rows instead.

Note: The reserved names COL and ROW must not be used as constant names.

Example:

DEF S=SWHE, T1='soft wheat', T2='Weichweizen';

- ASS statement

ASS statements are used to assign description text to unique description text keys. All access of SPEL programs to description text is effected via these by this description keys.

Syntax of text assignments:.

ASS K=*key*, T=*des* **[**,U=*unit***]**;

- or -

ASS K=*key*, T1=*des* **[**,T2=*des*, ... **]** **[**,U=*unit***]**;

- or -

ASS C=*ms.col.row*, T=*des* **[**,U=*unit***]**;

- or -

ASS C=*ms.col.row*, T1=*des* **[**,T2=*des*, ...**]** **[**, U=*unit***]**;

where:

Parameter K:

key Explicitly specified unique description key (up to 10 characters) enclosed in apostrophes.

Parameter C:

ms Combination of model area and table structure type
m specify one model area code such as 'S'
s specify 'C' for complementary table or 'B' for basic table

col Column code selection for table structure *ms*. Specify one of the following:
- a single code of up to 4 characters, e.g. SWHE;
- a sequence of codes separated by blanks, e.g. SWHE BARL MAIZ;
- a logical range of codes by specifying the lower and upper limit
  e.g. SWHE - OCRO;
- a combination of single codes, sequences and ranges,
  e.g. SWHE BARL - MAIZ OCRO

row Row code selection for table structure *ms*. For specifying rules see *col*.

Note: Specifying C=*ms.col.row* instead of K=*key* will implicitly define the related unique description key. In this case, the key will be internally built from model codes *ms*, *col* and *row* by concatenating them in this sequence. If *col* or *row* consists of sequences and/or ranges, each single ASS statement will generate several description entries, one for each resulting code combination.

Parameter T, T1, T2,.....:

des Description text to be assigned to the explicitly or implicitly specified key. Either the same text (T=*text*) or different text (T1=*text*, T2=....) may be assigned for the different languages (1, 2,...). The resulting text of up to 78 characters may be concatenated from literals and symbolic text constants in any sequence, e.g. 'Yield coefficient of' SWHE

Any sequence of consecutive blanks outside of text literals will result in one single blank. To concatenate named text constants without additional blanks, empty literals like NAME1''NAME2 may be used .

If the key is implicitly defined, symbolic text constants may also be referenced indirectly by using the reserved names COL and ROW. These special names are translated to symbolic text constant names whose name is the same as to the current column - resp. row code. e.g. to the row code MILK.

The suffixes '.c' resp. '.r' may be used in the DEF section to specify different text for column and row when the same code is used in the model for both, a column and a row. Any sequence of consecutive blanks outside of literals will result in one single blank.

Parameter UNIT:

unit Unit description text. The resulting text of up to 12 characters may be built from literals and symbolic text constants in any sequence. To assign unit description text in different languages, use symbolic text constants.

Example 1:

```
ASS C=SB.SWHE.LEVL, T='Main area soft wheat', U='1000 ha';
```

(is the same as)

```
ASS K='SBSWHELEVL', T='Main area soft wheat', U='1000 ha';
```

Example 2:

```
DEF S=HA3,  T='1000 ha';
DEF S=SWHE, T1='soft wheat',     T2='Weichweizen';
DEF S=DWHE, T1='durum wheat',    T2='Hartweizen';
DEF S=RYE,  T1='rye',            T2='Roggen';
DEF S=BARL, T1='barley',         T2='Gerste';
DEF S=LEVL, T1='activity level',T2='Prozessumfang';
....
ASS C=SB.SWHE-BARL.LEVL, T=LIN COL,  U=HA3;
```

The ASS statement will implicitly define descriptions for the keys 'SBSWHELEVL', 'SBDWHELEVL', 'SBRYE LEVL' and 'SBBARLLEVL' in two languages. The resulting description text for the key 'SBSWHELEVL' in language 1 is 'activity level soft wheat' and the resulting unit description is '1000 ha'.

Example 3:

```
DEF S=KGHD,    T1='kg/hd',          T2='kg/Ko';
DEF S=YIELD,   T1='yield',          T2='Ertrag',
DEF S=MILK.c,  T1='dairy cows',     T2='Milchkuehe';
DEF S=MILK.l,  T1='cows milk',       T2='Kuhmilch';
...
ASS C=SB.MILK.MILK, T:COL', 'YIELD LIN, U=KGHD
```

The ASS statement will implicitly define descriptions for the key 'SBMILKMILK' in two languages. The resulting description text in language 1 is 'dairy cows, yield cows milk' and the resulting unit description is 'kg/hd'.

General rules:

- Abbreviations: Underscored characters '_' may be used to mark optional abbreviation points inside words in literals like 'S_oft wheat'. Depending on the maximum length of the whole description allowed in a SPEL application program, the text will be printed either as 'Soft wheat' or 'S. wheat'.

- Accented characters, e.g. in French, German and Italian text. This special characters are not defined in the EBCDIC character encoding table of the IBM compatible mainframe. This characters are therefore paraphrased here as follows:

'a' with circumflex = 'a' preceded by '@c'
'a' with grave = 'a' preceded by '@g'
'a' with acute = 'a' preceded by '@a'
'a' with dieresis = 'a' preceded by '@d'
'A' with dieresis = 'A' preceded by '@d'
'a' with ring = 'a' preceded by '@r'
'A' with ring = 'A' preceded by '@r'
'c' with cedilla = 'c' preceded by '@z'
'e' with circumflex = 'e' preceded by '@c'
'e' with grave = 'e' preceded by '@g'
'e' with acute = 'e' preceded by '@a'
'E' with acute = 'E' preceded by '@a'
'e' with dieresis = 'e' preceded by '@d'
'i' with dieresis = 'i' preceded by '@d'
'n' with tilde = 'n' preceded by '@t'
'N' with tilde = 'N' preceded by '@t'
'o' with circumflex = 'o' preceded by '@c'
'o' with dieresis = 'o' preceded by '@d'
'O' with dieresis = 'O' preceded by '@d'
'u' with circumflex = 'u' preceded by '@c'
'u' with grave = 'u' preceded by '@g'
'u' with acute = 'u' preceded by '@a'
'u' with dieresis = 'u' preceded by '@d'
'U' with dieresis = 'U' preceded by '@d'

Since all these accented characters are part of the IBM PC character set, we offer a KEDIT-macro named ACCENT to translate from/to the paraphrased form. Please remember: DO NOT transfer the non-paraphrased (IBM-PC-) form to the IBM mainframe.

- Languages numbers used for SPEL:

  T1: English
  T2: German
  T3: French
  T4: Italian

- Table structure identifiers:

  EB: Integrated Supply & Demand Base Table structure
  SB: Supply-oriented Base Table structure
  DB: Demand-oriented Base Table structure
  SC: Supply-oriented Complementary Table structure
  DC: Demand-oriented Complementary Table structure

- Abbreviation always used without translation:

  PN: pure nutrient
  Reinnaehrstoff
  Substance nutritive pure
  puro nutriente

  NC: national currency
  Nationale Waehrung
  Monnaie nationale
  valuta nazionale

  NC85: national currency at constant prices (base year 1985)
  Nationale Waehrung in konst. Pr. (Basisjahr 1985) /
  Monnaie nationale ... prix constants (ann,e de base 1985) /
  valuta nazionale a prezzi costanti (anno base 1985)

- Abbreviation always used with translation:

  EAA: Economic Accounts for Agriculture
  LGR: Landwirtschaftliche Gesamtrechnung
  CEA: Comptes Economiques de l'Agriculture
  CEA: Conti economici dell'agricoltura
  PPS: Purchasing Power Standard
  KKS: Kauf-Kraft-Standard
  SPA: Standard de pouvoir d'achat
  PAS: potere di acquisto standard
  AWU: Annual Work Unit
  JAE: Jahres-Arbeitseinheit
  UTA: Unit, de travail annuelle
  ULA: unit... lavoro annue

Example of a DEF file

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--

 DEF S=KGHD,  T1='kg/hd', T2='kg/Ko', T3='kg/te', T4='kg/ca';
 DEF S=KGPNHD,T1='kg PN/hd', T2='kg RN/Ko', T3='kg PN/hd',T4='kg PN/hd';
 DEF S=BEEF.c,T1='ma_le ad_ult catt_le fatt_ening',
              T2='Mast erw_achsene mae_nnl. Rin_der',
              T3='bov_ins ad_ultes mâles à l'engr_aissement',
              T4='ingras_so vit_elloni';
 DEF S=BEEF.l,T1='beef',                    T2='Ri_nd_Fl_eisch',
              T3='vi_ande de boeuf',        T4='car_ne bov_ina';
 DEF S=HEIF,  T1='heif_ers',                T2='Fär_sen',
              T3='génisses',                T4='manze';
 DEF S=MILK.c,T1='d_airy cows',
              T2='Mi_lch_Kühe',
              T3='vach_es lait_ières',
              T4='vac_che da latte';
 DEF S=MANN,  T1='nitr_ogen f_rom ma_nure',
              T2='Stickst_off a_us Du_ng',
              T3='azote de fumier',
              T4='azoto da deiez_ioni';
 DEF S=MANP,  T1='phos_phate f_rom ma_nure',
              T2='Phos_phat a_us Du_ng',
              T3='phos_phate de fumier',
              T4='fosf_ato da deiez_ioni';
 DEF S=MANK,  T1='pota_ssium f_rom ma_nure',
              T2='Ka_li a_us Du_ng',
              T3='phos_phate de fumier',
              T4='pot_assio da deiez_ioni';
*
*  ...  (definitions of other symbolic constants, e.g. for MUTT)
*  ...
*
 ASS C=SB.HEIF.     BEEF,      T=COL', 'YIELD LIN,          U=KGHD;
 ASS C=SB.MILK-MUTT.MANN-MANK, T=COL', 'YIELD LIN,          U=KGPNHD;
*
----+----1----+----2----+----3----+----4----+----5----+----6----+----7--
```

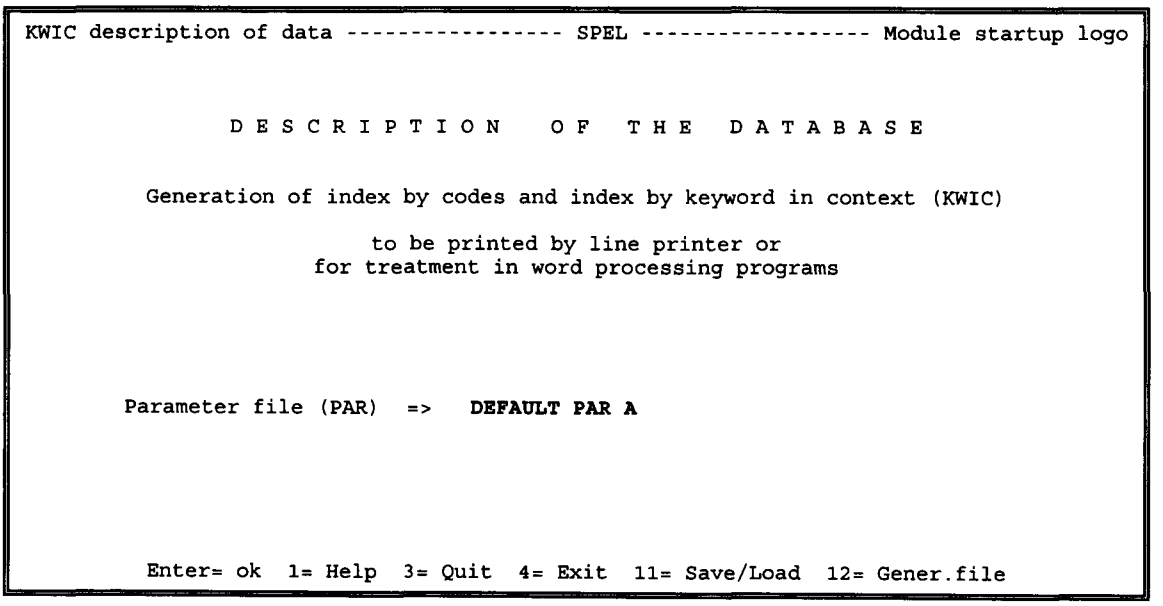
6.6.2. KWIDES program: generate tabulated data descriptions

The KWIDES program generates complete tabulated code descriptions for the SPEL database from the entries of DES input files. (Remember: The code description text originally defined in the DEF file is compiled by the DESCRI program and the result is stored in DES files.)

The output can be generated either for printing by general printers or for loading into word processing programs. The output consists of two parts, an index by table element codes and an index by keywords in context (KWIC). As explained in the description of the DEF files, abbreviation marks may be defined for the SPEL code descriptions. If defined, these abbreviations may or may not be fashioned so as to fit the whole description text of a code into a specified record length of the output file.

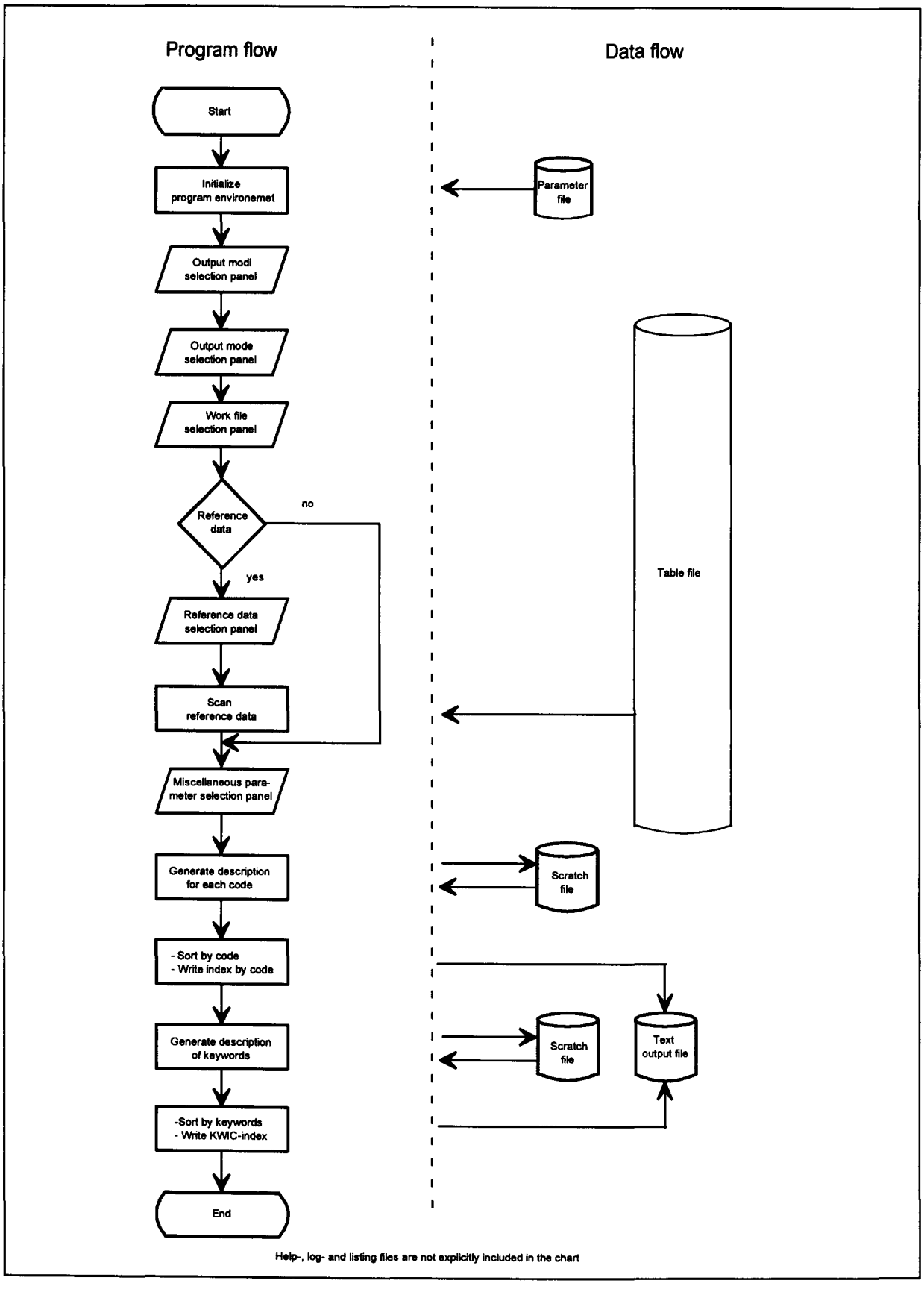
Index entries are generated either for all codes for which descriptions are found in the DES file or for all codes where related data exist in the SPEL database.

Module startup screen

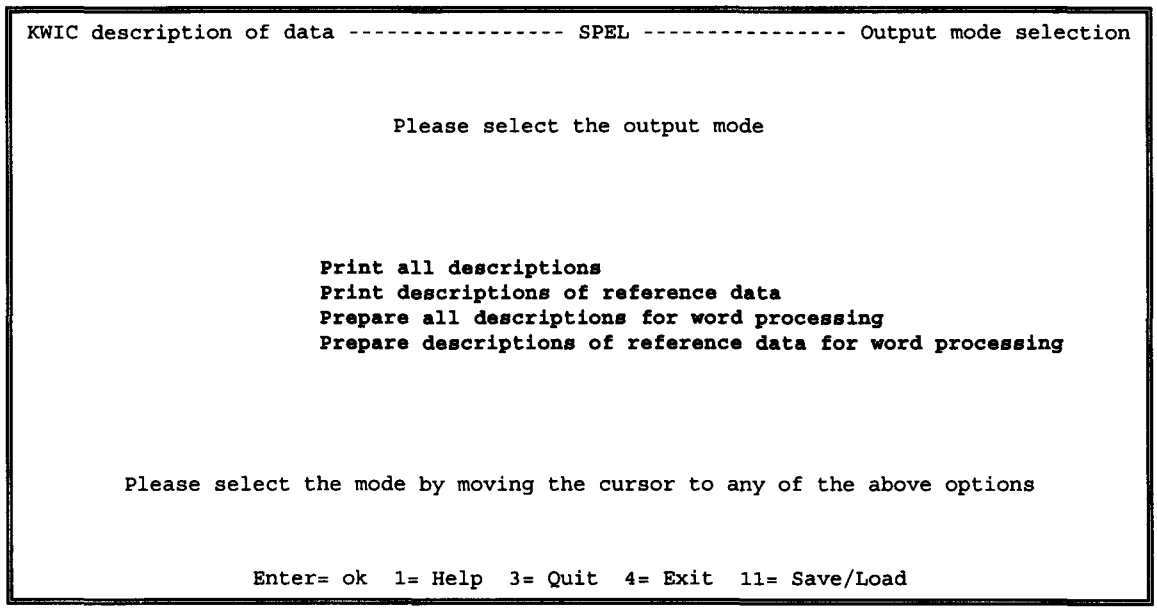


For detailed information, see chapter 'General programs'.

Figure 14: KWIDES program flow chart



Service selection screen



The output mode can be preselected by the 'Output mode' parameter in the parameter file.

Parameters:

Print all descriptions

Build a general print file from all descriptions found in the text input file.

Print descriptions of reference data

Build a general print file which describes reference data. An index entry will be generated for a table element, provided the element in at least one SPEL table of the reference data is non 'data missing'.

Prepare all descriptions for word processing

Build a file ready to load in a word processing program from all descriptions found in the text input file.

Prepare descriptions of reference data for word processing

Build a file ready to load in a word processing program which describes reference data. An index entry will be generated for a table element, provided the element in at least one SPEL table of the reference data is non 'data missing'.

Work file selection screen

```
KWIC description of data ---------- SPEL ---------- Work file selection

                          Please enter file names




Text input file (DES)      => ENGLISH DES F
Data reference file (TAB) => SPEL-SYS TAB F
Text output file (DOC)     => ENGLISH DOC C




            Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Text input file (DES)

Text descriptions of SPEL codes. This input file is a special application of the TAB format and is built by the DESCRI program. It contains the full descriptions for all elements of SPEL tables in a specific language (ENGLISH, GERMAN, FRENCH or ITALIAN).

Data reference file (TAB)

This input file of TAB format contains the database to describe. The 'reference data' will be read from this file.

(This parameter will not be displayed if the selected output mode does not require reference data.)

Text output file (PRN | DOC)

Output prepared for listings or word processing programs.

Remark: In the example above, 'Prepare all descriptions for word processing' was selected.

Reference data selection screen

```
KWIC description of data -------------- SPEL -------------- Reference data selection

                              Please select table keys


Region        ( 3 ch.) => E12

Sub-region    ( 2 ch.) => 00

Current year  ( 2 ch.) => 80 : 85

Periodicity   ( 2 ch.) => 00

Base year     ( 2 ch.) => NN

Type          ( 4 ch.) => BASB

Model area    ( 1 ch.) => S D


                  Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

This panel will be displayed if the selected output mode requires reference data.

An index entry will be generated for a table element, provided the element in at least one SPEL table of the reference data is non 'data missing'.

This panel will be displayed if the selected output mode require reference data.

Parameters:

Region

Region selection. Specify a single region code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

Sub-region

Sub-regions selection. Specify a single sub-region code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

Sub-regions are currently not used. The whole region has the sub-region code '00'.

Current year

Year selection. Specify a single year, a sequence of years, a numerical range or a combination of sequences and ranges.

Periodicity

Periodicity selection. Specify a single periodicity code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*'.

Periodicities are currently not used. The whole year has the code '00'.

Base year

Base year selection. Specify a single base year, a sequence of base years or an alphanumerical range or a combination of sequences and ranges.

For selecting SPEL tables of the ex-post period from the SPEL work file (TAB file), the code for the subkey base year will always be 'NN'.

Type

Type selection. Specify a single type code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

Model area

Model area selection. Specify a single model area code, a sequence of codes, an alphanumerical range, a combination of sequences and ranges or '*' .

For detailed information on code selection see chapter 'General program control'.

Miscellaneous parameter selection screen

```
KWIC description of data -------------- SPEL -------------- Miscellaneous parameters

                      Please select miscellaneous parameters




 Max. output record length in characters                => 95

 Exclude (ENGLISH/GERMAN/FRENCH/ITALIAN) trivial words => ENGLISH




                Enter= ok  1= Help  3= Quit  4= Exit  11= Save/Load
```

Parameters:

Max. output record length in characters

The descriptions read from the text input file are shortened to fit into output records of the specified length.

Exclude (ENGLISH/GERMAN/FRENCH/ITALIAN) trivial words

For each of the languages supported, tables of 'trivial words' (e.g : and, at, for, of, on...) are defined in the program. Specify the language of the input text file to ensure using the correct table for exclusion.

7. INDEX

ES Clasificación de las publicaciones de Eurostat

TEMA

1 Estadísticas generales (azul oscuro)
2 Economía y finanzas (violeta)
3 Población y condiciones sociales (amarillo)
4 Energía e industria (azul claro)
5 Agricultura, silvicultura y pesca (verde)
6 Comercio exterior y balanza de pagos (rojo)
7 Servicios y transportes (naranja)
8 Medio ambiente (turquesa)
9 Diversos (marrón)

SERIE

A Anuarios
B Coyuntura
C Cuentas, encuestas y estadísticas
D Estudios y análisis
E Métodos
F Estadísticas rápidas

GR Ταξινόμηση των δημοσιεύσεων της Eurostat

ΘΕΜΑ

1 Γενικές στατιστικές (βαθύ μπλε)
2 Οικονομία και δημοσιονομικά (βιολετί)
3 Πληθυσμός και κοινωνικές συνθήκες (κίτρινο)
4 Ενέργεια και βιομηχανία (μπλε)
5 Γεωργία, δάση και αλιεία (πράσινο)
6 Εξωτερικό εμπόριο και ισοζύγια πληρωμών (κόκκινο)
7 Υπηρεσίες και μεταφορές (πορτοκαλί)
8 Περιβάλλον (τουρκουάζ)
9 Διάφορα (καφέ)

ΣΕΙΡΑ

A Επετηρίδες
B Συγκυρία
C Λογαριασμοί, έρευνες και στατιστικές
D Μελέτες και αναλύσεις
E Μέθοδοι
F Ταχείες στατιστικές

IT Classificazione delle pubblicazioni dell'Eurostat

TEMA

1 Statistiche generali (blu)
2 Economia e finanze (viola)
3 Popolazione e condizioni sociali (giallo)
4 Energia e industria (azzurro)
5 Agricoltura, foreste e pesca (verde)
6 Commercio estero e bilancia dei pagamenti (rosso)
7 Servizi e trasporti (arancione)
8 Ambiente (turchese)
9 Diversi (marrone)

SERIE

A Annuari
B Tendenze congiunturali
C Conti, indagini e statistiche
D Studi e analisi
E Metodi
F Note rapide

DA Klassifikation af Eurostats publikationer

EMNE

1 Almene statistikker (mørkeblå)
2 Økonomi og finanser (violet)
3 Befolkning og sociale forhold (gul)
4 Energi og industri (blå)
5 Landbrug, skovbrug og fiskeri (grøn)
6 Udenrigshandel og betalingsbalancer (rød)
7 Tjenesteydelser og transport (orange)
8 Miljø (turkis)
9 Diverse statistikker (brun)

SERIE

A Årbøger
B Konjunkturoversigter
C Regnskaber, tællinger og statistikker
D Undersøgelser og analyser
E Metoder
F Ekspresoversigter

EN Classification of Eurostat publications

THEME

1 General statistics (midnight blue)
2 Economy and finance (violet)
3 Population and social conditions (yellow)
4 Energy and industry (blue)
5 Agriculture, forestry and fisheries (green)
6 External trade and balance of payments (red)
7 Services and transport (orange)
8 Environment (turquoise)
9 Miscellaneous (brown)

SERIES

A Yearbooks
B Short-term trends
C Accounts, surveys and statistics
D Studies and analyses
E Methods
F Rapid reports

NL Classificatie van de publikaties van Eurostat

ONDERWERP

1 Algemene statistiek (donkerblauw)
2 Economie en financiën (paars)
3 Bevolking en sociale voorwaarden (geel)
4 Energie en industrie (blauw)
5 Landbouw, bosbouw en visserij (groen)
6 Buitenlandse handel en betalingsbalansen (rood)
7 Diensten en vervoer (oranje)
8 Milieu (turkoois)
9 Diverse statistieken (bruin)

SERIE

A Jaarboeken
B Conjunctuur
C Rekeningen, enquêtes en statistieken
D Studies en analyses
E Methoden
F Spoedberichten

DE Gliederung der Veröffentlichungen von Eurostat

THEMENKREIS

1 Allgemeine Statistik (Dunkelblau)
2 Wirtschaft und Finanzen (Violett)
3 Bevölkerung und soziale Bedingungen (Gelb)
4 Energie und Industrie (Blau)
5 Land- und Forstwirtschaft, Fischerei (Grün)
6 Außenhandel und Zahlungsbilanz (Rot)
7 Dienstleistungen und Verkehr (Orange)
8 Umwelt (Türkis)
9 Verschiedenes (Braun)

REIHE

A Jahrbücher
B Konjunktur
C Konten, Erhebungen und Statistiken
D Studien und Analysen
E Methoden
F Schnellberichte

FR Classification des publications d'Eurostat

THÈME

1 Statistiques générales (bleu nuit)
2 Économie et finances (violet)
3 Population et conditions sociales (jaune)
4 Énergie et industrie (bleu)
5 Agriculture, sylviculture et pêche (vert)
6 Commerce extérieur et balance des paiements (rouge)
7 Services et transports (orange)
8 Environnement (turquoise)
9 Divers (brun)

SÉRIE

A Annuaires
B Conjoncture
C Comptes, enquêtes et statistiques
D Études et analyses
E Méthodes
F Statistiques rapides

PT Classificação das publicações do Eurostat

TEMA

1 Estatísticas gerais (azul escuro)
2 Economia e finanças (violeta)
3 População e condições sociais (amarelo)
4 Energia e indústria (azul)
5 Agricultura, silvicultura e pesca (verde)
6 Comércio externo e balança de pagamentos (vermelho)
7 Serviços e transportes (laranja)
8 Ambiente (turquesa)
9 Diversos (castanho)

SÉRIE

A Anuários
B Conjuntura
C Contas, inquéritos e estatísticas
D Estudos e análises
E Métodos
F Estatísticas rápidas

European Commission

SPEL system – Technical documentation (Rev. 1) – Vol. 1: Basics

Luxembourg: Office for Official Publications of the European Communities

1995 – 185 pp. – 21.0 × 29.7 cm

Theme 5: Agriculture, forestry and fisheries (green)

Series E: Methods

ISBN 92-826-9770-3

Price (excluding VAT) in Luxembourg: ECU 15 (Volume 1)
ECU 24 (Volumes 1 + 2)

The technical documentation of the SPEL system comprises two comprehensive volumes for users and operators working with the SPEL on-line system. It outlines the principal technical approach for the whole Spel system and points out the user interface flexibility and transparency. Transparency means that data can be traced back to the basic sources and underlying assumptions. Volume 1 describes the fundamental concepts. It begins with the system requirements, followed by an outline of formal structuring and coding, and concludes with the principles of data management and user interfaces. A supplementary chapter contains a description of general SPEL programs and utilities.